



Control basado en reconocimiento facial para un robot diferencial Facial recognition-based control for a differential robot

A. F. Arriaga-Morales ^{1,*}, J. Santiaguillo-Salinas ¹

División de Estudios de Posgrado, Universidad del Papaloapan, 68400, Loma Bonita, Oaxaca, México.

Resumen

En este trabajo se presenta el control de un robot móvil diferencial a partir del reconocimiento de un rostro objetivo. Para llevar a cabo el reconocimiento facial, se escogieron tres modelos de redes neuronales convolucionales (CNN's) pertenecientes al framework DeepFace: ArcFace, OpenFace y DeepID. Posteriormente, mediante un análisis comparativo de efectividad, velocidad y costo computacional, se seleccionó el modelo más adecuado para este trabajo. Las coordenadas del centroide del rostro, obtenidas del modelo elegido, son enviadas al robot diferencial para controlar su movimiento. Los resultados son validados experimentalmente utilizando una plataforma conformada por un robot diferencial Lego EV3 y un sistema de posicionamiento local OptiTrack.

Palabras Clave: Redes Neuronales Convolucionales, Reconocimiento Facial, Robot Móvil Diferencial, Control de Movimiento.

Abstract

This paper presents the control of a differential-drive mobile robot based on the recognition of a target face. To perform facial recognition, three convolutional neural networks (CNN's) models belonging to the DeepFace framework were chosen: ArcFace, OpenFace, and DeepID. Subsequently, through a comparative analysis of effectiveness, velocity and computational cost, the most appropriate model for this work was selected. The coordinates of the face centroid, obtained from the chosen model, are sent to a differential-drive robot to control its movement. The results are experimentally validated using a platform consisting of a Lego EV3 differential-drive robot and an OptiTrack local positioning system.

Keywords: Convolutional Neural Networks, Facial Recognition, Differential-drive Mobile Robot, Motion Control.

1. Introducción

La inteligencia artificial (IA), con sus diversas ramas como aprendizaje automático y aprendizaje profundo, se ha posicionado como una de las disciplinas más relevantes en ciencias de la computación, destacando por su amplia gama de aplicaciones. El interés por comprender y modelar la inteligencia humana ha impulsado a numerosos investigadores a explorar este campo.

Uno de los modelos más utilizados en aprendizaje profundo son las redes neuronales convolucionales (CNN), especialmente eficaces en tareas relacionadas con imágenes, donde han demostrado alto rendimiento. Su arquitectura multicapa combina convoluciones y funciones de activación no lineales, permitiendo extraer y procesar características visuales de forma progresiva Lopez Pacheco (2021).

Unas de las aplicaciones más comunes de las CNN's es la detección y reconocimiento facial. En Htet *et al.* (2021),

Srisuk y Ongkittikul (2017), Huang *et al.* (2022) y Herrera-Ollachica *et al.* (2024), se emplea una CNN para tareas en distintos contextos, como el control de acceso en sistemas de seguridad, la videovigilancia, la autenticación facial y la generación de bases de datos de imágenes. En Bonam *et al.* (2022) y Vyshnavi *et al.* (2024) utilizan distintas CNN's para desarrollar sistemas de asistencia en aulas escolares. De manera similar, en Lee *et al.* (2021) se construye un modelo basado en imágenes orientado a un futuro sistema de aula inteligente. Finalmente, en Firmansyah *et al.* (2023) se compara la precisión de tres CNN's con rostros de personas indonesas, señalando que los resultados podrían variar evaluando la precisión con rostros de otras razas.

Por su parte, la robótica ha experimentado un notable avance en las últimas décadas, motivado por el progreso tecnológico y la creciente demanda de automatización. Si bien la robótica y la IA son disciplinas distintas, han generado una sinergia fundamental entre ellas. Al integrar IA en los siste-

*Autor para correspondencia: afer.arriaga@gmail.com

Correo electrónico: afer.arriaga@gmail.com (Ana Fernanda Arriaga Morales), jsantiaguillo@outlook.com (Jesús Santiaguillo Salinas).

Historial del manuscrito: recibido el 25/09/2025, última versión-revisada recibida el 04/03/2026, aceptado el 05/03/2026, en línea (postprint) desde el 25/03/2026, publicado el 05/07/2026. **DOI:** <https://doi.org/10.29057/icbi.v14i27.16138>



mas robóticos, estos ya no solo perciben y actúan en su entorno, sino que también adquieren capacidades más avanzadas como el aprendizaje, la planificación y la toma de decisiones. Esta convergencia está dando lugar a sistemas autónomos cada vez más inteligentes y versátiles. Dicha integración se evidencia en diversos trabajos encontrados en la literatura. En Fujii *et al.* (2021), Cao *et al.* (2023) y Liang (2025) se utilizan redes neuronales para estimar incertidumbres en los modelos dinámicos de un robot submarino, un manipulador y robots móviles a ruedas, respectivamente. En Blum *et al.* (2021), Li y Li (2021) y Chang *et al.* (2024) se utiliza redes neuronales para planificar trayectorias o aprender las rutas mas cortas y seguras en entornos con obstáculos para la navegación de robots móviles. En Seghiri *et al.* (2022) y Mahmud *et al.* (2024) se usan redes neuronales profundas en robots diferenciales para lograr el seguimiento de trayectorias predefinidas, estimando los errores de seguimiento y generando las velocidades requeridas.

A diferencia de las propuestas citadas anteriormente que se centran en el uso de las CNN's para la estimación de dinámicas no conocidas o para la generación directa de trayectorias de un robot móvil, la principal contribución de este trabajo radica en la integración de una CNN para realizar reconocimiento facial con una estrategia de control para la navegación de un robot mediante una linealización por retroalimentación de estados. De esta forma original, se utiliza la precisión de una CNN para alimentar un esquema de control cinemático, garantizando el seguimiento del rostro sin depender exclusivamente del aprendizaje profundo para realizar la navegación del robot.

El trabajo se limitará a desarrollar e implementar un sistema de reconocimiento facial utilizando una cámara fija; el sistema será capaz de reconocer un rostro en particular y seguir el movimiento del mismo dentro del área de visión de la cámara. Se obtendrán las posiciones del centroide del rostro y se utilizarán para el cálculo de una estrategia para el control de movimiento de un robot diferencial Lego EV3. Posteriormente, se enviarán las señales correspondientes al robot para lograr su navegación.

Se probarán tres modelos de CNN pertenecientes al framework DeepFace: ArcFace, OpenFace y DeepID para la parte del reconocimiento facial. Se llevará a cabo un análisis comparativo considerando la velocidad de ejecución, el rango de precisión, la incidencia de falsos positivos y la fluidez del sistema. Con base en dichos resultados, se seleccionará el modelo más adecuado para implementarlo en la navegación del robot.

Este trabajo se organiza de la siguiente manera. En la Sección 2 se presenta una evaluación de las CNN's aplicadas a la detección, reconocimiento y seguimiento facial. La Sección 3 aborda el modelado y control del robot diferencial. La Sección 4 describe la plataforma experimental utilizada, así como los resultados obtenidos. Finalmente, en la Sección 5 se exponen las conclusiones.

2. Reconocimiento Facial

El reconocimiento facial se ha convertido en un tema de gran relevancia en la actualidad. Las primeras metodologías para la identificación de rostros se basaban en coordenadas de rasgos geométricos como los ojos, nariz y boca; sin embargo, se evidenció que dicha representación resultaba insuficiente para una identificación robusta. En respuesta, surgió el algoritmo

Eigenfaces, el cual identifica a las personas analizando el rostro como una imagen global, en lugar de medir rasgos sueltos como la distancia entre los ojos o la forma de la boca Turk y Pentland (1991). Posteriormente, fueron surgiendo otras metodologías o modelos empleando distintas técnicas llegando a lo que hoy conocemos como redes neuronales. Esta evolución ha sido clave para consolidar la identificación precisa de rostros como una herramienta fundamental.

En la literatura reciente se evidencia una clara tendencia hacia el uso del framework DeepFace en tareas de reconocimiento facial, tal como se destaca en Peng *et al.* (2019). Su versatilidad ha sido validada en aplicaciones de reconocimiento en vivo Bhanupriya *et al.* (2023) y en estudios comparativos que analizan la precisión de sus distintas arquitecturas internas Firmansyah *et al.* (2023). Asimismo, investigaciones más complejas, como la de Keerthana *et al.* (2025), implementan múltiples modelos de este framework para sistemas de monitoreo en tiempo real. En conjunto, estos trabajos reportan un rendimiento consistente y robusto, justificando la elección de esta herramienta para la presente investigación.

En este trabajo se presentan tres arquitecturas de CNN integradas en el framework DeepFace: ArcFace, OpenFace y DeepID. Cada una de ellas aborda el reconocimiento facial desde perspectivas distintas, aportando ventajas específicas en términos de efectividad, velocidad y costo computacional. ArcFace emplea una red profunda basada en ResNet-100, un margen angular en el espacio del coseno y un vector de características que es una representación numérica del rostro de una persona, lo que permite generar representaciones altamente discriminativas y mejorar la separación entre identidades. OpenFace, inspirado en FaceNet, genera vectores de características compactos de 128 dimensiones y utiliza la función de pérdida Triplet Loss, lo que facilita la comparación directa entre rostros a través de distancias euclidianas, siendo además, un enfoque eficiente y ligero. Finalmente, DeepID fue uno de los primeros modelos en demostrar la eficacia de las redes convolucionales profundas, extrayendo descriptores de alta dimensión que posteriormente se clasifican mediante funciones Softmax o máquinas de soporte vectorial.

El esquema de entrenamiento de los modelos se presenta en la Figura 1, tomando como ejemplo ArcFace.

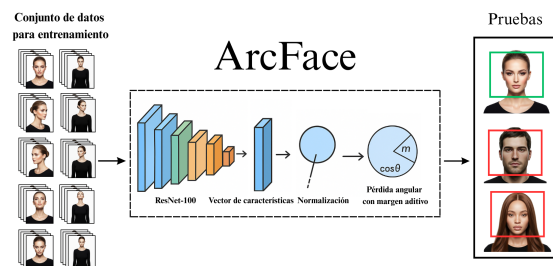


Figura 1: Esquema de entrenamiento con ArcFace

Las pruebas de las tres CNN se realizaron en una laptop Lenovo IdeaPad 3 15ALC6 equipada con un procesador AMD Ryzen 5 5500U con Radeon Graphics 2.10 GHz, 12 GB de RAM y SO Windows 11, utilizando una cámara web HD. El conjunto de datos consistió en 100 fotografías de un rostro objetivo en 10 posiciones distintas: frente a la cámara en primer

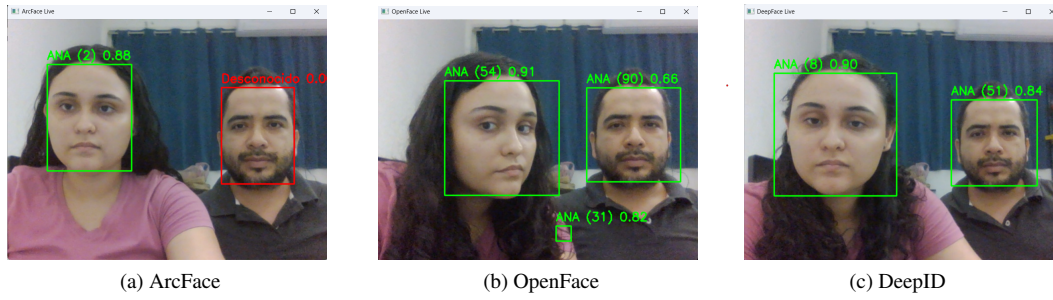


Figura 2: Comparativa de las CNN's

plano, de perfil izquierdo, de perfil derecho, y nuevamente de frente mirando hacia arriba y hacia abajo. Posteriormente, se repitieron las mismas cinco posiciones a una mayor distancia de la cámara. Las fotografías fueron tomadas dentro de laboratorio y aunque no se tuvieron variaciones bruscas en las condiciones de luz, se utilizó un fondo complejo que incluían diversos objetos presentes en el laboratorio y de esta forma tener condiciones de operación más realistas. Se utilizaron las 100 fotografías para el entrenamiento y se procedió a la evaluación mediante video en tiempo real, incluyendo la presencia de distintos rostros no objetivos para evaluar los falsos positivos.

Para la comparación de los modelos se consideró la velocidad de ejecución, que se refiere al tiempo de entrenamiento y ejecución inicial de las CNN, el rango de precisión alcanzado, es decir, el porcentaje de coincidencia del rostro detectado con el rostro objetivo, la presencia de falsos positivos y fluidez del sistema durante la captura en tiempo real. La fluidez del sistema hace referencia a la estabilidad del centroide del rostro, esta se evaluó mediante la desviación estándar (σ) del desplazamiento del centroide entre frames del video. Un valor bajo en σ indica una fluidez alta (seguimiento suave y estable) y un valor alto una fluidez baja (seguimiento con variaciones abruptas). Los resultados obtenidos se muestran en la Tabla 1 y algunas capturas de las pruebas realizadas donde se aprecian la precisión y presencia de falsos positivos se pueden ver en la Figura 2.

Tabla 1: Tabla comparativa de CNN's

CNN	Velocidad	Precisión	Falsos positivos	Fluidez
ArcFace	00:51:40	72-85 %	No	Alta
OpenFace	00:24:07	63-96 %	Si	Alta
DeepID	04:48:22	66-94 %	Si	Baja

A partir del análisis realizado, la CNN seleccionada para lograr el control de movimiento del robot diferencial a partir del reconocimiento facial es ArcFace, dado que obtuvo los mejores resultados en términos de detección y velocidad. Además, no presentó falsos positivos y mostró un desempeño eficiente considerando las especificaciones del equipo utilizado.

3. Modelado y control del robot diferencial

Dentro de la robótica móvil, una de las plataformas robóticas más utilizadas por los investigadores debido a su simplicidad

de construcción y control es el robot móvil diferencial. Su mecanismo de tracción consta de dos ruedas motrices que se accionan independientemente generando su locomoción. Para entender el movimiento del robot diferencial se emplea el modelado cinemático, el cual describe el movimiento del robot a partir de sus velocidades Siciliano y Khatib (2008).

Considere el robot diferencial de la Figura 1, donde $\xi = [x, y, \theta]^T$ describe la pose del robot, con $\{x, y\}$ la posición del punto medio del eje de las ruedas y θ el ángulo de dirección. Es sabido que, el modelo cinemático del robot diferencial presenta una singularidad al considerar como salida de control el punto $\{x, y\}$. Para evitar tal singularidad en el modelo, se utiliza un punto $\rho = [\rho_x, \rho_y]^T$ en la parte frontal del robot como salida de control Santiaguillo y Aranda (2017). De esta forma, el modelo cinemático del robot diferencial está dado por

$$\begin{bmatrix} \dot{\rho}_x \\ \dot{\rho}_y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = A(\theta, \ell) \begin{bmatrix} v \\ w \end{bmatrix} \quad (1)$$

donde ℓ es la distancia del punto medio del eje de las ruedas al punto frontal, $A(\theta, \ell)$ es la matriz de desacoplamiento no singular ya que $\det A(\theta, \ell) = \ell$, v y w son las velocidades lineal y angular, respectivamente.

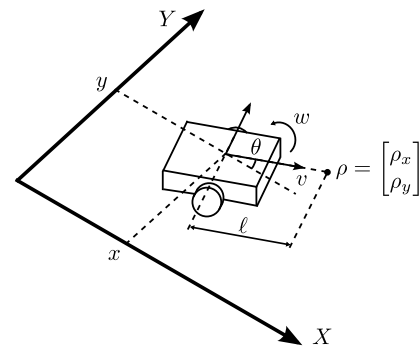


Figura 3: Esquema del robot diferencial

Para el control del robot proponemos una estrategia de linealización por retroalimentación de estados dada por

$$\begin{bmatrix} v \\ w \end{bmatrix} = A^{-1}(\theta, \ell) [k_p(\rho - m(t)) + \dot{m}(t)] \quad (2)$$

con $A^{-1}(\theta, \ell)$ la inversa de la matriz de desacoplamiento, $m(t)$ la trayectoria deseada proveniente del reconocimiento y seguimiento de rostro, $\dot{m}(t)$ la velocidad de la trayectoria deseada estimada mediante diferenciación numérica y k_p la ganancia de control. El esquema de control se presenta en la Figura 4.

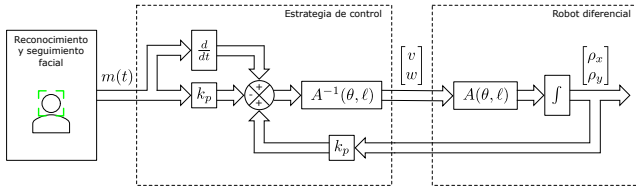


Figura 4: Esquema de la estrategia de control

4. Experimentación

4.1. Plataforma experimental

Para implementar la CNN en el control de movimiento de un robot diferencial, se cuenta con la plataforma experimental mostrada en la Figura 5. La plataforma está formada por un sistema de reconocimiento y seguimiento facial, un sistema de control, un robot diferencial Lego EV3, un sistema de posicionamiento local (LPS) y un sistema de comunicación.



Figura 5: Esquema de la plataforma experimental

El sistema de reconocimiento y seguimiento facial está formado por una laptop dedicada y una cámara web HD. Con la cámara web se obtiene el video en tiempo real de un área en donde estarán en movimiento distintas personas; este video es enviado a la laptop dedicada con SO Windows, en donde se ejecuta un programa en Python con el modelo de la CNN entrenado con el rostro objetivo. El programa reconoce solamente el rostro objetivo y una vez ubicado en el video, realiza el seguimiento del mismo, obteniendo la posición del centroide del rostro en píxeles, como se muestra en la Figura 6. La posición del rostro es enviada al sistema de control.

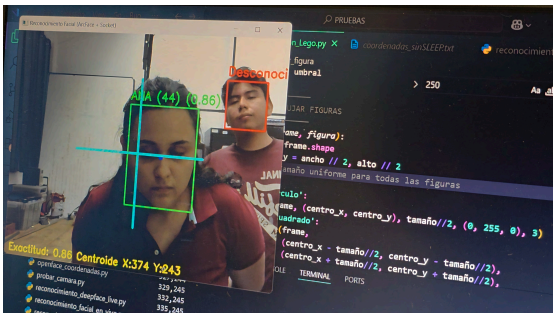


Figura 6: Imagen del programa en ejecución

El sistema de control se implementa en una computadora con SO Windows, en la cual se ejecuta un programa en Python

que recibe la posición del rostro objetivo en píxeles y realiza una transformación de coordenadas para pasar de la posición en píxeles a posición en metros, correspondientes al área de trabajo del robot. Esta transformación está dada por

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} \left(x_p - \frac{W_p}{2}\right) / x_{ppm} \\ -\left(y_p - \frac{H_p}{2}\right) / y_{ppm} \end{bmatrix} \quad (3)$$

donde $\{x_m, y_m\}$ es la trayectoria deseada $m(t)$ en metros, $\{x_p, y_p\}$ es la trayectoria deseada en píxeles, $\{W_p, H_p\}$ es la resolución de la imagen en píxeles (640x640) y $\{x_{ppm}, y_{ppm}\}$ son los píxeles por metro en ancho y alto (320x320), respectivamente. Estos últimos valores se calibraron considerando un factor fijo basado en el área de trabajo. Con la posición del rostro en metros y la posición del robot, se realiza el cálculo de la estrategia de control, obteniendo las velocidades lineal y angular del robot diferencial. Estas velocidades son enviadas al robot.

El robot diferencial utilizado fue construido con el kit de robótica de Lego Mindstorm EV3. El robot Lego EV3 es controlado mediante el Brick o ladrillo, como se le conoce comúnmente, el cual es una microcomputadora a la cual se le montó un SO basado en Linux, en el cual se ejecuta un programa en MicroPython, donde recibe las velocidades lineales y angulares del robot y mediante el uso de las librerías de Lego son convertidas a velocidades angulares para cada motor.

La posición y orientación del robot Lego EV3 se obtienen a partir del LPS conformado por un conjunto de cámaras infrarrojas OptiTrack y una computadora con el software Motive. Las cámaras detectan el centroide de un conjunto de marcadores infrarrojos colocados sobre el robot, enviando dicha información al software Motive, que se ejecuta en una computadora con SO Windows. En Motive se calcula la posición y orientación del robot y se envía bajo un protocolo de red NatNet.

El sistema de comunicación está formado por una red inalámbrica local levantada mediante un router, comunicando mediante esta red a todos los componentes de la plataforma.

4.2. Resultados experimentales

A continuación se muestran los resultados obtenidos de 2 experimentos teniendo varios rostros en pantalla. Para cada experimento, el rostro objetivo realizó diferentes trayectorias, una cruz y un triángulo. En los experimentos, las trayectorias se mostraban en pantalla, como se aprecia en la Figura 6, esto con la finalidad de que la persona con el rostro objetivo se guiara al tratar de realizar el movimiento con el rostro. Los parámetros utilizados en los experimentos son $\ell = 0,1m$ y $k_p = 1,2$. El valor de k_p se ajustó de manera experimental para obtener el mejor resultado posible en el seguimiento, mientras que el valor de ℓ se eligió a partir de las medidas físicas del robot.

Experimento 1. A continuación se muestran los resultados obtenidos del experimento donde el rostro objetivo realizó una trayectoria en forma de cruz. La Figura 7 muestra los desplazamientos del robot Lego EV3 en el área de trabajo. En la Figura 8 se muestran los errores de posición del robot. Se puede ver que los errores tratan de converger a cero; sin embargo, se presentan oscilaciones. Estas oscilaciones se deben al movimiento irregular del centroide del rostro objetivo, ya que dicho movimiento no es suave y tiene cambios repentinos de velocidad. Las señales de control, es decir, las velocidades lineal y angular enviadas al robot se muestran en la Figura 9.

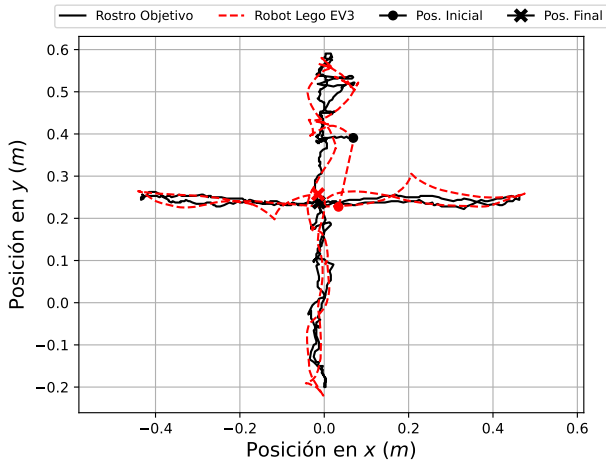


Figura 7: Trayectorias del robot Lego EV3.

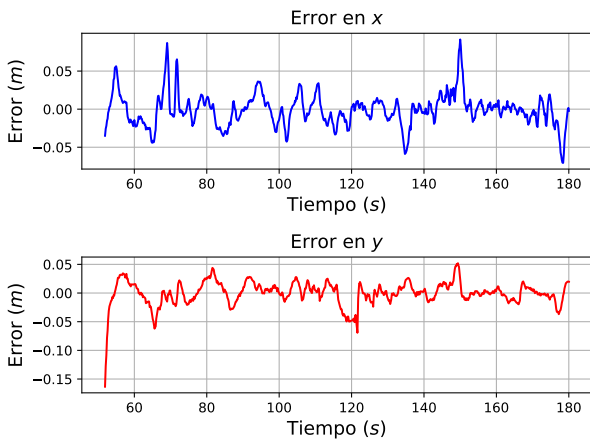


Figura 8: Errores de posición del robot Lego EV3.

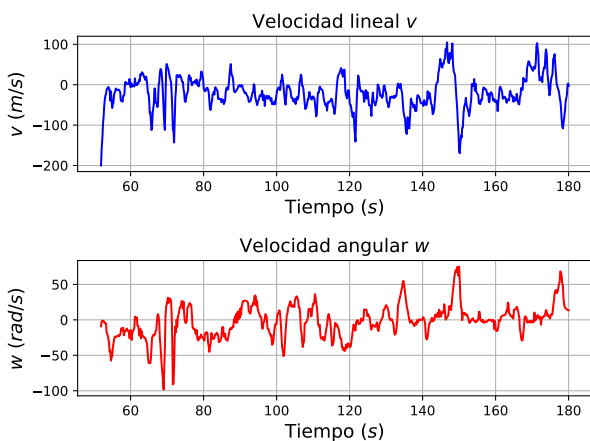


Figura 9: Señales de control del robot Lego EV3.

Experimento 2. A continuación se presentan las gráficas con los resultados obtenidos del segundo experimento, donde el rostro objetivo realizó una trayectoria en forma de triángulo. La Figura 10 muestra los desplazamientos del robot Lego EV3

en el área de trabajo. En la Figura 11 se muestran los errores de posición del robot. Al igual que en el experimento 1, los errores se mantienen oscilando dentro de una región cercana a cero, teniendo errores máximos por debajo de los 6 cm. Las señales de control enviadas al robot se muestran en la Figura 12.

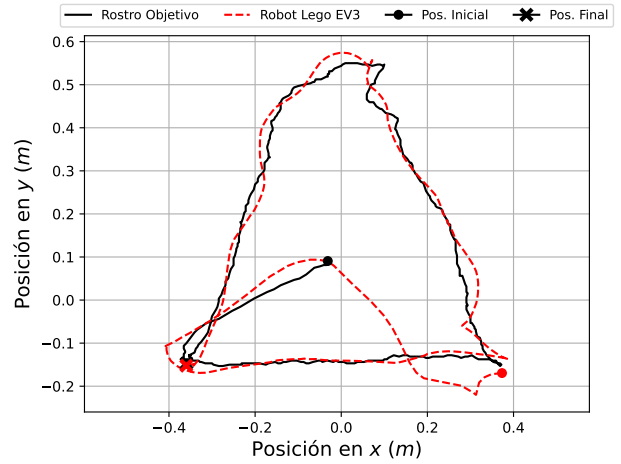


Figura 10: Trayectorias del robot Lego EV3.

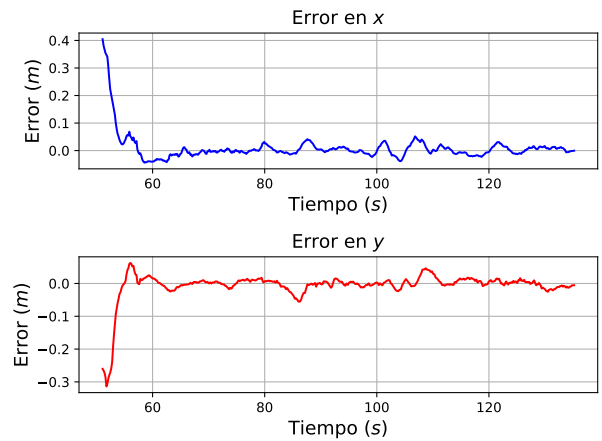


Figura 11: Errores de posición del robot Lego EV3.

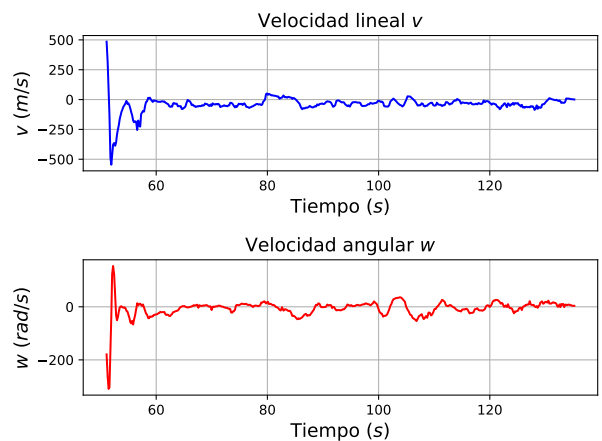


Figura 12: Señales de control del robot Lego EV3.

Es posible observar que ambos experimentos empiezan en un tiempo de 51.4 segundos, esto debido a como se comentó en la Sección 2, el tiempo que tarda en ejecutarse la CNN elegida (ArcFace) es de 51.4 segundos. Una vez ejecutada la CNN, el experimento se realiza sin problemática alguna.

Como se menciona en la Sección 3, el cálculo de la velocidad deseada o velocidad del rostro objetivo, se calcula mediante una derivada numérica dada por

$$\begin{bmatrix} \dot{x}_m \\ \dot{y}_m \end{bmatrix} = \begin{bmatrix} \frac{x_m(t) - x_m(t-h)}{h} \\ \frac{y_m(t) - y_m(t-h)}{h} \end{bmatrix} \quad (4)$$

con h el tamaño de paso, variable entre 0.15 y 0.18 segundos de acuerdo al tiempo de ejecución en cada iteración del programa en Python. Una vez calculada la derivada numérica, se elimina el ruido mediante un filtro promedio. Estos cálculos se realizan ya que no es posible obtener de manera directa las velocidades del rostro objetivo.

5. Conclusiones

En este trabajo se presentó el análisis de desempeño de tres CNN's integradas en el framework DeepFace: ArcFace, OpenFace y DeepID para lograr la detección, reconocimiento y seguimiento de un rostro objetivo. Se determinó que ArcFace tiene un mejor desempeño sin falsos positivos, una fluidez alta, con buena precisión y velocidad de ejecución.

Con la CNN elegida, se diseñó e implementó un sistema de control para lograr que un robot diferencial se desplazara dentro de un área de trabajo, siguiendo el movimiento o trayectorias del centroide del rostro objetivo.

Los resultados obtenidos fueron validados de forma experimental sobre una plataforma conformada por un robot diferencial Lego EV3 y un sistema de posicionamiento local OptiTrack, mediante el cual se obtenía la posición y orientación del robot dentro del área de trabajo. Las gráficas presentadas demuestran el buen desempeño de la CNN en conjunto con la estrategia de control, logrando que el robot Lego EV3 realice de forma exitosa el seguimiento del rostro. Sin embargo, es posible apreciar que, debido al movimiento irregular del centroide del rostro y el cálculo de su velocidad de forma numérica, aunado a dinámicas no modeladas como deslizamiento de las llantas y fricción, se presentan errores que oscilan en un rango por debajo de los 6 cm y que en algunos instantes, donde el movimiento es más brusco, pueden tener picos de hasta 8 cm.

Como trabajo a futuro se pretende mejorar el sistema de reconocimiento facial realizando pruebas con otros algoritmos de redes neuronales, por ejemplo Yolo. Además, realizar pruebas de detección, reconocimiento y seguimiento de rostros utilizando una cámara abordo del robot, con la finalidad de aplicar el sistema a tareas de búsqueda y rescate, vigilancia, entre otras.

Referencias

Bhanupriya, M., Kirubakaran, N., y Jegadeeshwari, P. (2023). Emotiontracker: Real-time facial emotion detection with opencv and deepface. En *2023 International Conference on Data Science, Agents Artificial Intelligence (ICDSAAI)*, pp. 1–4.

- Blum, T., Paillet, G., y Masawat, W. Yoshida, K. (2021). Segvisr!: development of a robot's neural visuomotor and planning system for lunar exploration. *Advanced Robotics*, 35(21-22):1359–1373.
- Bonam, J., Burra, L. R., Godavarthi, R. S. V., Jagabattula, D., Eda, S., y Gogulamudi, S. (2022). *Intelligent Computing and Applications*, volumen 315, capítulo Multiple Face Recognition System Using OpenFace, pp. 339–349. Springer, Singapore.
- Cao, M., Zhou, X., y Yunpeng, J. (2023). Robot motion planning based on improved rrt algorithm and rbf neural network sliding. *IEEE Access*, 11:121295–121305.
- Chang, L., Yuan, H., Wang, T., Mai, T., y Yang, L. (2024). A convex optimization based differentially driven mobile robot planner for crowd navigation. (*2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*), pp. 1531–1538.
- Firmansyah, A., Fabrianti Kusumasari, T., y Novrizal Alam, E. (2023). Comparison of face recognition accuracy of arcface, facenet and facenet512 models on deepface framework. *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, pp. 535–539.
- Fujii, S., Tahara, J., Zhang, F., Koike, M., Ohta, Y., y Watanabe, Y. (2021). Motion control of deep sea vehicle 'otohime': modeling with neural network. *Advanced Robotics*, 35(24):1500–1512.
- Herrera-Ollachica, D. A., Asiedu-Asante, B. K., e Imamura, H. (2024). Autonomous uav implementation for facial recognition and tracking in gps-denied environments. *IEEE Access*, 12:119464–119487.
- Htet, A., V. Bobkov, A., y Lin Tun, N. (2021). Face detection in real time live video using yolo algorithm based on vgg16 convolutional neural network. *2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 697–702.
- Huang, J. Tang, S., Teng, Z., Zhang, Y., y Zhou, X. (2022). Face tracking and recognition in video moving images based on convolutional neural network. *Multimedia Tools and Applications*, 81:33771–33784.
- Keerthana, P., Sabu, M., Shaji, P., Johnson, T., y George, R. A. (2025). Intelligent campus monitoring and tracking. En *2025 11th International Conference on Smart Computing and Communications (ICSCC)*, pp. 1–6.
- Lee, G. C., Lee, Y., y Chiang, C. (2021). Low-resolution face recognition in multi-person indoor environments using convolutional neural networks. *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1629–1633.
- Li, X. y Li, M. (2021). The direction analysis on trajectory of fast neural network learning robot. *IEEE Access*, 9:125580–125589.
- Liang, Y. (2025). Robot trajectory tracking control based on neural networks and sliding mode control. *IEEE Access*, 13:96740–96757.
- Lopez Pacheco, M. A. (2021). *Redes neuronales convolucionales para el modelado de sistemas no lineales con aplicacion al monitoreo de dano estructural*. Tesis doctoral, CINVESTAV.
- Mahmud, S. A., Kamarulariffin, A., Ibrahim, A. M., y Mohideen, A. J. H. (2024). Advancements and challenges in mobile robot navigation: A comprehensive review of algorithms and potential for self-learning approaches. *Journal of Intelligent Robotic Systems (2024)*, pp. 110–120.
- Peng, L., Xin, Z., y Ping, G. (2019). Design and implementation of remote deepface model face recognition system based on sbrio fpga platform and nb-iot module. En *2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, pp. 505–509.
- Santiagoullo, J. y Aranda, E. (2017). Containment problem with time-varying formation and collision avoidance for multiagent systems. *International Journal of Advanced Robotic Systems*, 14(3).
- Seghiri, S. E., Mansouri, N., y Chemori, A. (2022). Implementation of sarl* algorithm for a differential drive robot in a gazebo crowded simulation environment. *2022 2nd International Conference on Advanced Electrical Engineering (ICAEE)*, pp. 1–6.
- Siciliano, B. y Khatib, O., editores (2008). *Springer Handbook of Robotics*. Springer Berlin, Heidelberg, 1st edición.
- Srisuk, S. y Ongkittikul, S. (2017). Robust face recognition based on weighted deepface. *2017 International Electrical Engineering Congress (iEECON)*, pp. 1–4.
- Turk, M. y Pentland, A. (1991). Face recognition using eigenfaces. En *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591.
- Vyshnavi, M. K., Shaik, S., Kumar, M. S., Praveen, M., y Potti, S. (2024). Facial recognition based attendance system using opencv. *The Journal of Computational Science and Engineering*, 2:1–10.