

Smart video reproduction system based on a convolutional neural network (CNN)

Leonardo Martínez-Espíritu* Jesús Morales-Valdez*

* *Colegio de Ciencia y Tecnología, Universidad Autónoma de la Ciudad de México, Calzada Ermita Iztapalapa No. 4163, Col. Lomas de Zaragoza, Alcaldía Iztapalapa, C.P. 09620, CDMX, México, (e-mail: leonardo.martinez.espiritu@alumnos.uacm.edu.mx, jesus.morales@uacm.edu.mx).*

Resumen: Recent advances in computing and technological development have significantly enhanced the importance of algorithms such as neural networks, deep learning, and artificial intelligence in practical applications. Indeed, innovations like the Raspberry Pi and similar devices have facilitated neural network programming. This work details the development and implementation of a smart video reproduction system capable of recognizing various hand gestures, which are interpreted as instructions via a convolutional neural network (CNN). Using a presence sensor (PIR), the system determines when to activate. The CNN also enables real-time classification of content from a webcam, ensuring that the played videos align with user preferences. The integration of the PIR sensor and CNN presents an innovative approach for automated video playback management, enhancing user-system interaction and optimizing the viewing experience. The system efficiently responds to user presence and personalizes content based on real-time analysis. All functionalities are programmed on a Raspberry Pi model 4 computer via python software. The results demonstrate the effectiveness of the proposed approach, highlighting the potential of presence detection, content classification, and adaptive technologies based on computer vision.

Palabras clave: convolutional neural network, image classification, open CV, tensorflow, raspberry PI.

1. INTRODUCCIÓN

El avance en la computación y el desarrollo tecnológico en los últimos años ha permitido la implementación de algoritmos de aprendizaje automático. En esta nueva era, algoritmos como las redes neuronales, aprendizaje profundo e inteligencia artificial han retomado una importancia significativa.

Una de las propiedades más destacadas de las redes neuronales es su naturaleza adaptativa, lo que significa que adquieren conocimiento del entorno mediante un proceso conocido como aprendizaje (Sjöberg et al., 1994). Se ha documentado que las redes neuronales poseen una amplia gama de aplicaciones, produciendo consistentemente resultados satisfactorios en diversos campos. Por ejemplo para tareas de estimación, identificación y control (Narendra and Parthasarathy, 1990), diagnóstico de daño basado en el reconocimiento de patrones (Sohn et al., 2003; Farrar et al., 2001), entre otras. Por otra parte, el aprendizaje profundo o deep learning es un conjunto avanzado de algoritmos que emplea un aprendizaje multinivel para representar datos o características de manera jerárquica. Un ejemplo particular de estos algoritmos son las redes

neuronales convolucionales, las cuales consisten en capas convolucionales y capas de submuestreo alternadas, seguidas de una capa completamente conectada (Schmidhuber, 2015). Derivado de las propiedades de la convolución en el procesamiento de imágenes, estas redes se han especializado en resolver problemas de clasificación de imágenes (Li et al., 2014; Lawrence et al., 1997), reconocimiento de voz (Nassif et al., 2019) y otros tipos de señales (Morales-Valdez et al., 2020).

En la tarea de clasificación de imágenes, a diferencia de las técnicas basadas en visión por computadora, como el método de reconocimiento basado en colores, que es intuitivo y fácil de entender para el usuario, esta técnica se caracteriza por ser compatible con cámaras y dispositivos convencionales, lo que facilita las tareas de reconocimiento y clasificación de imágenes. No obstante, también presenta ciertas desventajas importantes, como su sensibilidad a cambios de iluminación y sombras. Requiere buena iluminación para un rendimiento óptimo, lo cual puede ser complicado en ciertos entornos no controlados, y es menos eficaz en condiciones de baja visibilidad para el dispositivo que captura las imágenes (Lamberti and Camastra, 2011). Por el contrario, el reconocimiento basado en aprendizaje profundo se emplea para aprender representaciones complejas y abstractas de datos visuales. Esta técnica es capaz de automatizar la extracción de características y

* Agradecimiento al Colegio de Ciencia y Tecnología de la Universidad Autónoma de la Ciudad de México por el apoyo otorgado a través de los proyectos UACM CCYT-2023-INI-02.

la toma de decisiones basadas en grandes volúmenes de datos (Oudah et al., 2020).

Comparando las opciones, el método de reconocimiento de colores sería ineficaz en un museo debido a los cambios de iluminación y diferentes tonalidades de piel. Las técnicas basadas en movimiento y profundidad también se verían afectadas por la falta de hardware adecuado. Por ello, optamos por el método de aprendizaje profundo utilizando Mediapipe, basado en una CNN con varios modelos preentrenados de Tensorflow, como "Mediapipe Hands Database" o "Hand Landmarker", que simplifican el problema de grandes cantidades de datos y el proceso de entrenamiento, haciendo su uso más accesible.

En este trabajo se presenta la implementación de una red neuronal convolucional para el desarrollo de un sistema de reproducción de video inteligente, conformado en dos etapas. La primera de ellas consta de un algoritmo de presencia basado en un sensor pasivo infrarrojo (PIR, por sus siglas en inglés de passive infrared sensor) que permite activar el sistema de reproducción de video cuando un usuario se aproxima a una posición definida previamente. En seguida se incorporan los algoritmos para el reconocimiento y clasificación de imágenes mediante una red neuronal convolucional (CNN, por sus siglas en inglés de convolutional neural network). La tarea de esta red es interpretar indicaciones a partir de la cantidad de dedos mostrados en una mano frente a una cámara web de bajo costo. A partir de las imágenes generadas por cualquier configuración creada por los cinco dedos de la mano, se puede elegir el video de su agrado. El sistema es capaz de contar los dedos de las manos a partir de un algoritmo de clasificación, sin importar que mano se use, o que dedos de coloquen frente a una cámara web. Este desarrollo se encuentra implementado en un dispositivo Raspberry Pi modelo 4, equipada con 8 GB de memoria RAM. Para validar el desempeño del sistema de reproducción inteligente, este será implementado en el "Sitio de Memoria" ubicado en la Universidad Autónoma de la Ciudad de México, plantel Casa Libertad". Los resultados obtenidos en diversas pruebas validan la fiabilidad de la propuesta. Cabe señalar que un resultado previo se encuentra implementado en el Museo Comunitario de Santiago Acahualtepec, ubicado en la Alcaldía Iztapalapa, CDMX.

El resto del artículo está organizado como sigue: la sección 2 describe la arquitectura de red neuronal convolucional empleada, desde la etapa del entrenamiento, testing, y las consideraciones correspondientes. El desarrollo experimental de la celda se describe en la Sección 3, al igual que la evaluación experimental. Finalmente, las conclusiones y observaciones finales se proporcionan en la Sección 4.

2. RED NEURONAL CONVOLUCIONAL (CNN)

La metodología para la reproducción inteligente de videos mediante el reconocimiento de imágenes consta esencialmente de una red neuronal convolucional que tiene la ventaja de ser robusta para compensar diversas configuraciones, giros, falta de iluminación, entre otras perturbaciones en las imágenes. Por ejemplo, si una imagen es girada 180° de su posición original, una red neuronal simple

no podría identificar la imagen original, mientras que la red neuronal convolucional puede identificarla satisfactoriamente. Adicionalmente, la operación de convolución es una propiedad importante que permite reducir el número de operaciones y el tiempo de cálculo, convirtiéndolo en un método rápido en comparación con una red neuronal tradicional. El segundo paso es diseñar un algoritmo de clasificación utilizando CNN. A continuación se explica cada uno de etapas desarrolladas.

2.1 Arquitectura de la CNN

La arquitectura CNN propuesta en este trabajo consta de dos capas convolucionales con su respectiva capa de submuestreo, seguidas de una capa completamente conectada, que conduce a la salida $\hat{F}(t)$. Esta estructura se puede ver en la Fig.1. Las capas convolucionales consisten en una colección de mapas de características generados por la convolución entre los filtros $\gamma^{(\ell)}$ y la salida de la capa anterior $\zeta^{(\ell-1)}$, donde ℓ indica la capa actual. Los mapas de características se generan de la siguiente manera:

$$\chi^{(\ell)} = \zeta^{(\ell-1)} \otimes \gamma^{(\ell)} \quad (1)$$

Después de generar cada mapa de características, se aplica una función de activación $\mathbb{C}ReLU$ como:

$$\zeta^{(\ell)} = \mathbb{C}ReLU(\chi^{(\ell)}) \quad (2)$$

Esta función de activación se define como:

$$\mathbb{C}ReLU(x) = \text{ReLU}(x_{\Re}) + i\text{ReLU}(x_{\Im}) \quad (3)$$

$$x = x_{\Re} + ix_{\Im} \in \mathbb{C}$$

donde la función de activación de la Unidad Lineal Rectificada (ReLU) es una operación que toma el valor máximo de sus dos argumentos y lo genera, es decir, $f(x) = \max(0, x)$. Por otro lado, la capa de submuestreo consta de la función máximo por magnitud, que ayuda a reducir los datos a través del CNN. Esta operación se define como:

$$\zeta^{(\ell)} = \arg \max_{x \in \text{set}} M(x) \quad (4)$$

donde

$$M(x) = |x| \quad M : \mathbb{C} \rightarrow \Re \quad (5)$$

Finalmente, la salida de la CNN define como:

$$\hat{F}(t) = \Gamma^T \zeta \quad (6)$$

donde, ζ es el vector concatenado de todos los mapas de características de las capas convolucionales y Γ son los pesos sinápticos de la capa de salida. Para el *entrenamiento* de la CNN, se utiliza el algoritmo de "regla delta", que permite actualizar los parámetros de la red, como filtros y pesos sinápticos.

2.2 Entrenamiento de CNN para la identificación.

El entrenamiento se lleva a cabo utilizando el algoritmo de Retro Propagación (BP, por sus siglas en inglés de Back Propagation) (Rumelhart et al., 1986; Fukuoka et al., 1998). Para la implementación de la red neuronal convolucional, el algoritmo se puede simplificar a través del pseudo-código presentado a continuación:

En este algoritmo, N_t representa la cantidad total de datos utilizados en el proceso. La inicialización de los

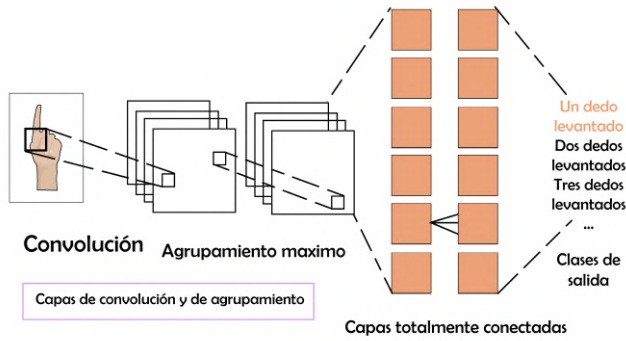


Fig. 1. Arquitectura de la red neuronal convolucional

Algorithm 1 CNN entrenamiento

```

repeat
  Etapa de avance
  Generación de vector de entrada  $\hat{x}$ 
  for  $\ell \in$  Número de capas do
    Cálculo de mapas de características
     $\chi^{(\ell)} = \zeta^{(\ell-1)} \otimes \gamma^{(\ell)}$ 
    Cálculo de  $\zeta^{(\ell)} = \arg \max_{x \in \text{set}} M(x)$ 
  end for
   $\hat{F}(t) = \Gamma^T \zeta$  ▷ CNN salida
  Retropropagación
   $e = \hat{F} - F$  ▷ Error de estimación
   $J = e^2$  ▷ Función de costo a minimizar
  for  $\ell \in$  Número de capas do
    Cálculo de  $\frac{\partial J}{\partial \Gamma}, \frac{\partial J}{\partial \gamma^{(\ell)}}$ 
    Actualización de hiperparámetros
     $\Gamma(k+1) = \Gamma(k) - \eta \frac{\partial J}{\partial \Gamma}$ 
     $\gamma^{(\ell)}(k+1) = \gamma^{(\ell)}(k) - \eta \frac{\partial J}{\partial \Gamma}$ 
  end for
until  $j = N_t$ 

```

parámetros se elige aleatoriamente dentro del rango $[-1, 1]$ para las capas completamente conectadas y los filtros de las capas convolucionales están dentro del rango $\lambda_i^{(\ell)} \in [-\frac{1}{\sqrt{j}}, \frac{1}{\sqrt{j}}]$, donde j es el tamaño de la entrada en esa capa. Los parámetros se actualizan empleando la función de costo (7)

$$J(q) = \frac{1}{2} \varepsilon^2(q) = [\hat{z}(q) - z(q)]^2 \quad (7)$$

Para la capa completamente conectada, la actualización de pesos es:

$$W(q+1) = W(q) - \zeta \frac{\partial J}{\partial W} \quad (8)$$

donde ζ es la tasa de aprendizaje.

2.3 Metodología para clasificación de imágenes

En esta tarea, la CNN está diseñada como clasificador (CNN-C). La nueva CNN-C tienen 1 salida para cada clase de datos a clasificar, brindando información relativa e una evaluación global, e informa si la imagen es compatible con alguna dentro de la base de datos.

Después de que la CNN converge y obtiene un resultado, estos pasan por el clasificador *softmax*, que calcula las "probabilidades" para todas las clases. La suma de las salidas del clasificador deben ser equivalente a 1. La función softmax se define como

$$p_i = \frac{\exp(\hat{y}_i)}{\sum_j^\theta \exp(\hat{y}_j)} \quad (9)$$

donde θ es el número total de clases y p_i representa la probabilidad predicha para cada clase. La arquitectura para la CNN como clasificador es la siguiente: 2 capas convolucionales *CL1* y *CL3* con filtros $r = 5$ por capa de tamaño $Lf_1 = Lf_3 = 3$, dos capas desubmuestreo *S2* y *S4* con una reducción $s^{(2)} = s^{(4)} = 2$. Dos capas completamente conectadas *F5* y *F6* con $LL = 75$ neuronas ocultas en la capa *F5*. Hay cinco clases entonces $\theta = 5$. La tasa de aprendizaje para la capa convolucional es 0,0001 y para ambas capas completamente conectadas se establece en 0,007.

Considerando que cada grupo de datos ha sido etiquetado, definimos $Y(j)$ como la probabilidad verdadera de la clase proporcionada por los datos. Aquí, se debe tener en cuenta que Y en general contiene solo el valor 1 en la clase verdadera y ceros en el resto, por lo que la suma de sus valores debe ser igual a 1. Definiendo la la entropía cruzada para la CNN como

$$J(Y, p) = - \sum_j Y_j \log(p_j) \quad (10)$$

Podemos obtener su gradiente para cada una de las salidas de CNN de la siguiente manera:

$$\frac{\partial J}{\partial \hat{y}_j} = p_j - Y_j \quad (11)$$

que es una expresión muy simple y elegante. Esta expresión, como puede verse, solo disminuye la pérdida cuando la salida corresponde a la clase etiquetada.

3. VALIDACIÓN EXPERIMENTAL

Con el objetivo de validar la red neuronal propuesta, se desarrolló un sistema inteligente de reproducción de video. En este sistema, el enfoque principal fue utilizar los movimientos de los dedos de la mano del usuario como método de selección de contenido.

El desarrollo está compuesto por una Raspberry Pi 4 Model B, que cuenta con 8 GB de memoria RAM y un procesador Broadcom BCM2711 Quad-core Cortex-A72 (ARM v8) con arquitectura de 64 bits. Además, ofrece 40 pines GPIO y múltiples mejoras respecto a modelos anteriores. Entre sus especificaciones destacan el soporte para dos pantallas 4K simultáneas, gracias a sus dos puertos micro HDMI, así como la inclusión de dos puertos USB 3.0, dos puertos USB 2.0 y un puerto Ethernet Gigabit. Asimismo, es compatible con un puerto USB tipo C para la alimentación eléctrica. También se usó una cámara web con una resolución de 0.9Mpx a 1080px x 720px dando una calidad de vídeo en HD, un módulo PIR con un rango de detección de cuatro metros de distancia y de tamaño compacto (2.45 cm x 3.26 cm x 2.4 cm). Además, se integró una bocina con entrada auxiliar de 3.5 mm para mejorar la calidad del audio

y garantizar que el usuario pueda escuchar los videos de manera adecuada. Para obtener un mejor marco de referencia sobre el montaje físico, consulte la Fig. 2.

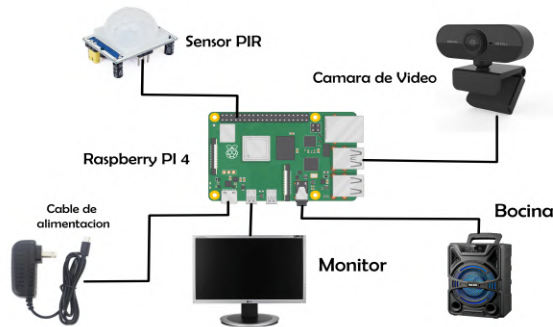


Fig. 2. Conexiones de la Raspberry Pi 4

3.1 Programación en la Raspberry

Para la implementación de este sistema basado en una red neuronal convolucional, primero fue necesario instalar el sistema operativo "Raspberry Pi OS (64-bit)" (anteriormente conocido como "Raspbian"), el cual está basado en Debian Linux. Este sistema operativo está optimizado y adaptado para funcionar de manera eficiente en la arquitectura ARM, que es la base de la Raspberry Pi, un SBC (Single-Board Computer, por sus siglas en inglés). Para llevar a cabo la programación, es esencial contar con las siguientes librerías:

- **OpenCV:** Facilita el procesamiento de imágenes y la captura de video. Captura fotogramas desde la cámara, convierte las imágenes a diferentes formatos y muestra los resultados procesados de manera eficiente.
- **MediaPipe:** Esta herramienta detecta y sigue características corporales, como manos, rostros y poses. En la aplicación desarrollada, su función principal es identificar los puntos clave de las manos para determinar con precisión la posición de los dedos.
- **Subprocess:** Concede la creación y gestión de procesos del sistema. Lanza y controla la reproducción de videos con el reproductor.
- **GPIOZero:** Simplifica la interacción con componentes de hardware GPIO en la Raspberry Pi y detecta movimiento por medio del sensor PIR.
- **Psutil:** Proporciona herramientas para el monitoreo y gestión de procesos del sistema.
- **Tensorflow:** Facilita la creación de modelos de aprendizaje máquina (ML, por sus siglas en inglés de Machine Learning) que pueden ejecutarse en cualquier entorno.

Además, para la reproducción de vídeo e imagen se instalaron los siguientes softwares:

- **Feh:** Es un visor de imágenes capaz de mostrar imágenes en un entorno de ventana X11, lo que lo convierte en una herramienta muy útil para scripts y procesos de automatización gracias a su gran flexibilidad.

- **MPV:** Es un reproductor multimedia versátil basado en MPlayer y mplayer2. Soporta una amplia variedad de formatos multimedia.

Con la intención de facilitar la comprensión del algoritmo desarrollado, la Fig. 3 presenta cada una de las etapas de la programación realizada, descrita en forma de diagrama de flujo. Además, se incluyó una imagen introductoria que

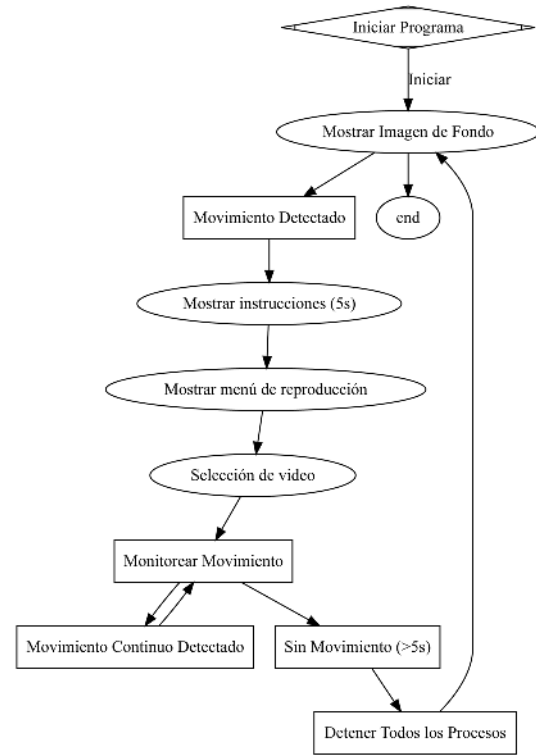


Fig. 3. Diagrama de flujo del programa desarrollado

guía al usuario en el uso del sistema, seguida del menú con los videos de prueba utilizados para la demostración, como se muestra en la Fig. 4.



Fig. 4. Instrucciones para el uso adecuado por parte del usuario.

Para evaluar el sistema de reproducción de vídeos propuesto, se realizaron 100 pruebas, de las cuales 85 fueron exitosas. Los errores en el resto de las pruebas se debieron principalmente a la resolución de la cámara y la distancia de los usuarios frente a esta; a mayor distancia, la red

no lograba identificar correctamente los objetos. Además, las limitaciones del procesador de la Raspberry Pi, aunque eficiente en su categoría, no se comparan con un Intel Core i5, lo que afectó el procesamiento de imágenes en algunos casos. No obstante, su bajo costo lo hace una opción prometedora para aplicaciones prácticas, como en museos. La Tabla 1 muestra el costo total de implementación del sistema.

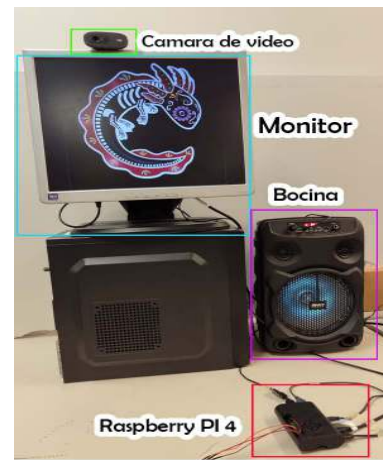
Tabla 1. Costos de los equipos

Equipo	Costo
KIT Raspberry Pi 4 MODELO B - 8GB con carcasa, ventilador, disipadores, cargador.	2,504 MXN
Cámara WEB 720p	350 MXN
Sensor PIR	35 MXN
Monitor	1,500MXN
	4,389 MXN

La Fig. 5 muestra las conexiones físicas del prototipo experimental del sistema inteligente de reproducción de video, desarrollado en el Laboratorio de Innovación, Investigación y Desarrollo Tecnológico (LIIDET) de la Universidad Autónoma de la Ciudad de México, en el plantel Casa Libertad. Este sistema se instalará en el espacio de memorias llamado "Las celdas de Casa Libertad". Cabe destacar que una versión preliminar del sistema ya está instalada y en funcionamiento en el "Museo Comunitario de Santiago Acahualtepec", ubicado en la Segunda Ampliación, Santiago Acahualtepec, Alcaldía Iztapalapa, Ciudad de México. En el siguiente enlace: <https://www.youtube.com/watch?v=Qh3kEhDdEpQ>, se puede observar el desarrollo mencionado anteriormente, en particular entre los minutos 1:04 y 1:06. Por otro lado, como se observa en la Fig. 6, el usuario debe acercarse al área de detección del sensor PIR para activar el sistema. A continuación, el monitor muestra dos imágenes en 10 segundos, que proporcionan las instrucciones de uso y el menú para seleccionar el video deseado. Una vez activado el sistema y mostradas las instrucciones al usuario a través del programa establecidos en la Fig. 4). El siguiente paso es que el usuario realice el gesto deseado frente a la cámara para que el video seleccionado se reproduzca automáticamente. Además, se ha añadido una opción para cambiar la selección al presentar un puño cerrado, como se muestra en la Fig. 7, donde un usuario se encuentra interactuando con el sistema.

4. CONCLUSIÓN

En este trabajo se ha presentado el desarrollo e implementación de un sistema inteligente de reproducción de videos, el cual tiene la capacidad de interpretar diferentes gestos a partir de las manos. El sistema no solo responde de manera eficiente al diseño de un clasificador de imágenes mediante una red neuronal convolucional, sino que también personaliza el contenido en función de análisis en tiempo real, lo que representa un avance significativo en la personalización de sistemas multimedia. El bajo costo de este desarrollo lo hace una tecnología prometedora para aplicaciones en museos, sitios de memoria y otros espacios que requieren equipamiento audiovisual. Sin embargo, aunque los resultados son favorables, los



(a) Configuración experimental del sistema de reproducción de video



(b) Sistema de reproducción de video con sensor PIR integrado

Fig. 5. Conexiones del Sistema de reproducción de video
autores reconocen que se deben realizar investigaciones más amplias en esta dirección. Por ejemplo, es notorio que el desempeño del clasificador se degrada cuando el usuario se ubica a una distancia considerable de la cámara, especialmente en rangos de hasta 20 metros, lo que provoca leves complicaciones en el sistema. Esto se debe principalmente a la baja resolución y características limitadas de la cámara comercial utilizada, ya que a mayor distancia, la red neuronal convolucional (CNN) encuentra más dificultades para clasificar correctamente las imágenes. Además, se ha observado que el rendimiento de la Raspberry Pi 4 no es suficiente para procesar eficientemente las imágenes, lo que provoca retrasos y errores en la clasificación de las instrucciones.

AGRADECIMIENTOS

El primer autor agradece a la Universidad Autónoma de la Ciudad de México (UACM) por la oportunidad brindada para realizar sus estudios de ingeniería.

REFERENCIAS

Farrar, C.R., Doebling, S.W., and Nix, D.A. (2001). Vibration-based structural damage identification. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1778), 131-149.



(a) Usuario acercándose al área de detección del sensor PIR



(a) Usuario realizando selección de video a través de los dedos



(b) Instrucciones y menú de usuario



(b) Sistema detectando cuatro dedos levantados, y enseguida un puño cerrado



(c) Vídeo reproduciéndose

Fig. 6. Primera etapa: detección de presencia a través del sensor PIR

Fukuoka, Y., Matsuki, H., Minamitani, H., and Ishida, A. (1998). A modified back-propagation method to avoid false local minima. *Neural Networks*, 11(6), 1059–1072.

Lamberti, L. and Camastra, F. (2011). Real-time hand gesture recognition using a color glove. In *Image Analysis and Processing-ICIAP 2011: 16th International Conference, Ravenna, Italy, September 14-16, 2011, Proceedings, Part I 16*, 365–373. Springer.

Lawrence, S., Giles, C.L., Tsoi, A.C., and Back, A.D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98–113.

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D.D., and Chen, M. (2014). Medical image classification with convolutional neural network. In *2014 13th international conference on control automation robotics & vision (ICARCV)*, 844–848. IEEE.

Morales-Valdez, J., Lopez-Pacheco, M., and Yu, W. (2020). Automated damage location for building structures using the hysteretic model and frequency domain neural networks. *Structural Control and Health Monitoring*, 27(9), e2584.

Narendra, K.S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1(1), 4–27.

Nassif, A.B., Shahin, I., Attili, I., Azzeh, M., and Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE access*, 7, 19143–19165.

Fig. 7. Usuario interactuando con el sistema

Oudah, M., Al-Naji, A., and Chahl, J. (2020). Hand gesture recognition based on computer vision: a review of techniques. *Journal of Imaging*, 6(8), 73.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.

Sjöberg, J., Hjalmarsson, H., and Ljung, L. (1994). Neural networks in system identification. *IFAC Proceedings Volumes*, 27(8), 359–382.

Sohn, H., Farrar, C.R., Hemez, F.M., Shunk, D.D., Stinemates, D.W., Nadler, B.R., and Czarnecki, J.J. (2003). A review of structural health monitoring literature: 1996–2001. *Los Alamos National Laboratory, USA*, 1, 16.