

Implementación de un control de movimiento para dispositivos robóticos móviles utilizando identificación de gestos faciales y orientación de cabeza por medio de visión de computadora

GERARDO ANDRES FUENTES BÁMACA

Universidad del Valle de Guatemala
Facultad de Ingeniería
Carrera Ingeniería Mecatrónica
Presentación Proyecto de Graduación
Guatemala 14 de junio de 2024



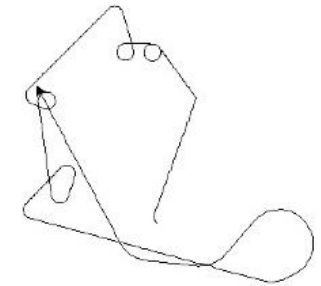
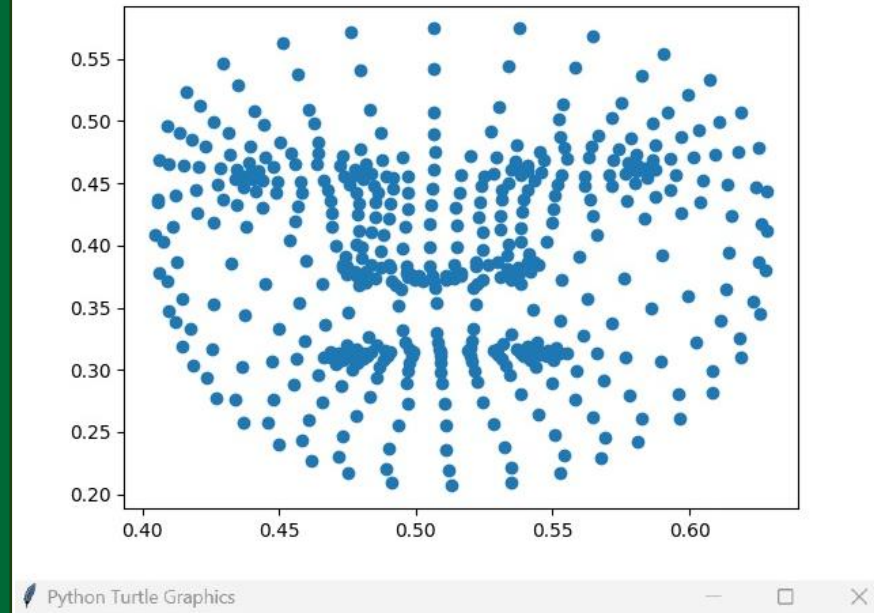
Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Implementación de un control de movimiento para dispositivos robóticos móviles utilizando identificación de gestos faciales y orientación de cabeza por medio de visión de computadora

GERARDO ANDRES FUENTES BÁMACA

Universidad del Valle de Guatemala
Facultad de Ingeniería
Carrera Ingeniería Mecatrónica
Presentación Proyecto de Graduación
Guatemala 14 de junio de 2024



Resumen

- Prototipo de control alternativo para plataformas móviles.
- Implementación de aprendizaje de máquina para reconocer las posiciones/gestos de la cabeza.

Resumen

- Integración de diversas herramientas para crear una interfaz completa de asistencia a usuarios.
- Utilización de tecnología desarrollada internacionalmente y dentro de la UVG.

Objetivos

General:

Desarrollar un sistema de visión por computadora para reconocimiento de gestos y orientación de la cabeza para el control de un agente robótico móvil.

Específicos:

- Investigar y seleccionar el hardware y software disponible para la aplicación de visión por computadora.
- Implementar un algoritmo de visión por computadora para reconocimiento de gestos y movimiento de la cabeza.

Objetivos

- Implementar un algoritmo de traducción de los gestos y movimientos de la cabeza a comandos de control para el agente robótico.
- Validar el correcto funcionamiento del sistema de reconocimiento y control por medio de simulaciones computarizadas.
- Validar el correcto funcionamiento del sistema de reconocimiento y control por medio de plataformas robóticas móviles.

Introducción

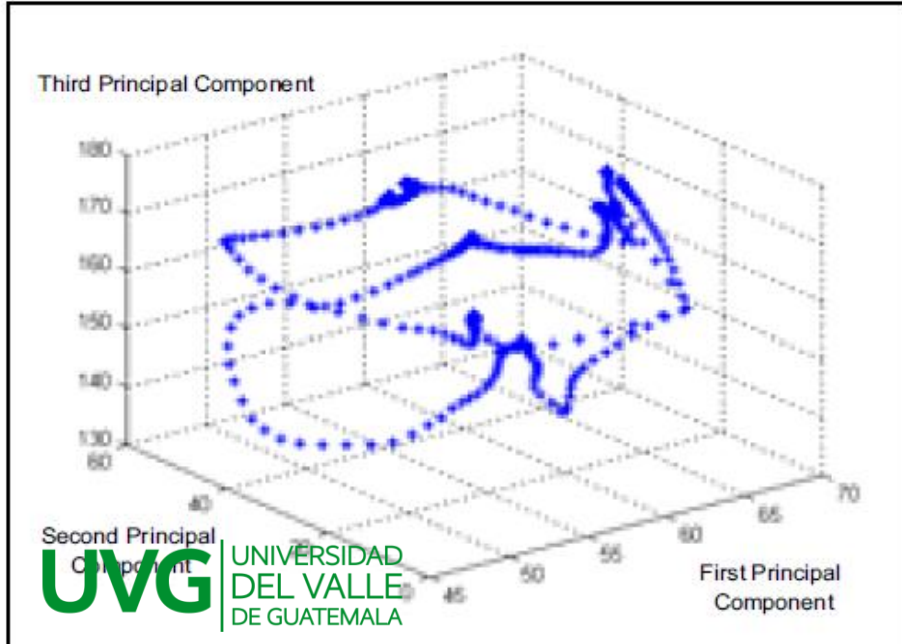


Introducción

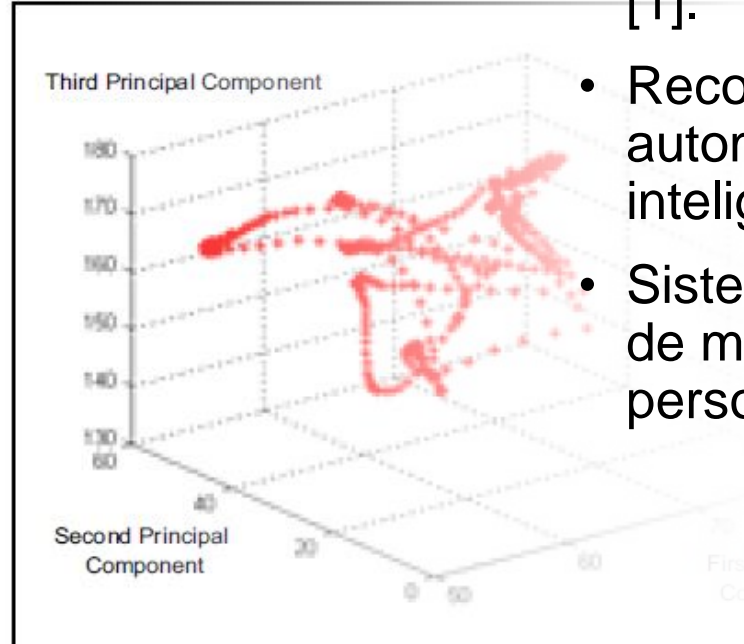
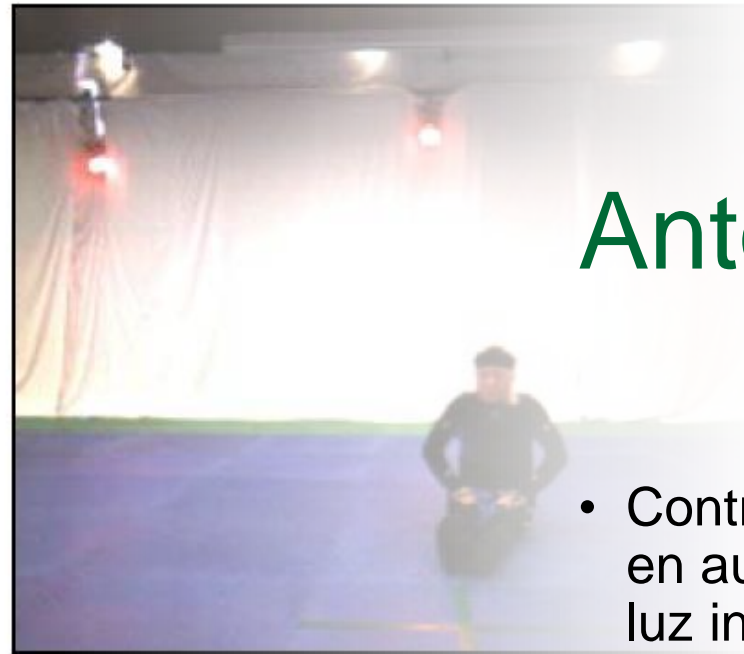
Este proyecto tiene como objetivo principal crear un prototipo de sistema de asistencia a usuarios para controlar el movimiento de una plataforma móvil con posiciones/gestos de la cabeza.

Fue necesario integrar diferentes herramientas que estuvieran disponibles para la mayoría del público en general y algunos únicamente dentro de la UVG, para lograr demostraciones en simuladores y a nivel físico.





Walking

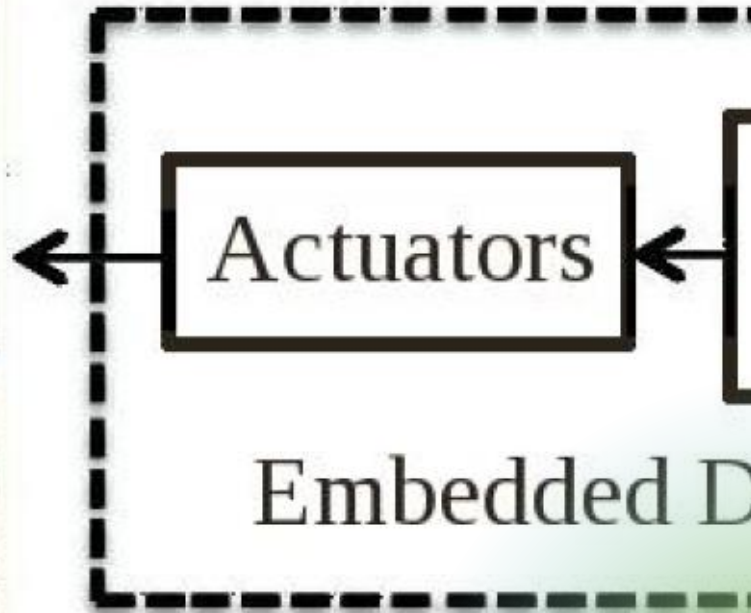


Sitting on the floor

Antecedentes

- Control de gestos para su uso en automóviles, por medio de luz infrarroja cercana (NIR) [1].
- Reconocimiento de gestos automático para interacciones inteligentes humano-robot [2].
- Sistema de reconocimiento de manos y rostro para personas ciegas [3].

Kinect



- Sistema de reconocimiento de gestos utilizando la cámara de gestos [4]
- Control de una silla de ruedas usando una agrupación de k-medias adaptativas de las poses de la cabeza [5].
- Mediapipe [6].

Justificación

- Asistencia a personas con extremidades inhabilitadas.
- Utilización de robots móviles abre paso para una futura aplicación con plataformas móviles como sillas de ruedas.
- Evitar el uso de tecnología invasiva para el cuerpo humano .



Alcance

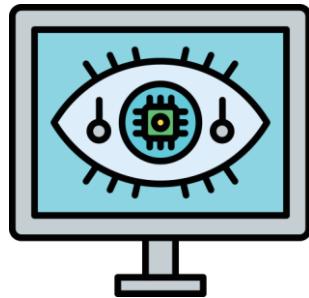
- Demostración de visión de computadora aplicada a agentes robóticos simples por medio de reconocimiento de gestos, espacios cerrados y laboratorios de universidades libres de obstáculos que puedan dañar la integridad del robot.
- En cuanto a la utilización, aún se requiere software específico, como Python, Webots y el sistema operativo de Windows, aún no es multiplataforma .

-
- Este sistema aún tiene ciertas deficiencias cuando la luz trasera de la imagen capturada es mayor a la luz frontal, dado que utiliza un modelo de aprendizaje predictivo y no uno generativo.
 - El sistema es capaz de capturar miles de datos en minutos, por lo que la limitación de datos depende de la capacidad de almacenamiento del dispositivo en uso.

Marco teórico



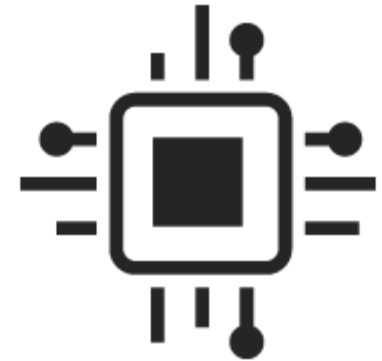
Redes Neuronales [7]



Visión de computadora [8]



Protocolos de comunicación [9]



Sistemas operativos en tiempo real [10]

Métodos

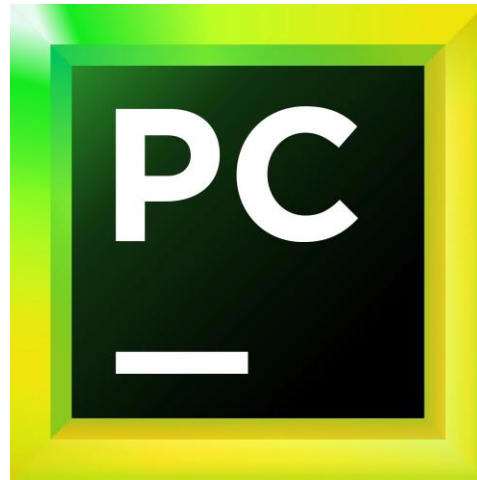
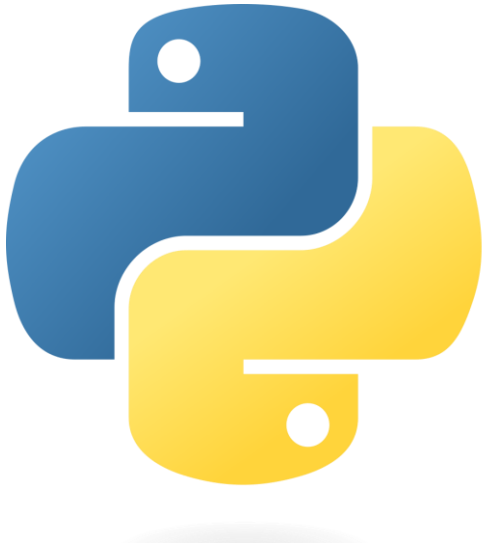


Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Selección de hardware y software

- Software de fuente abierta

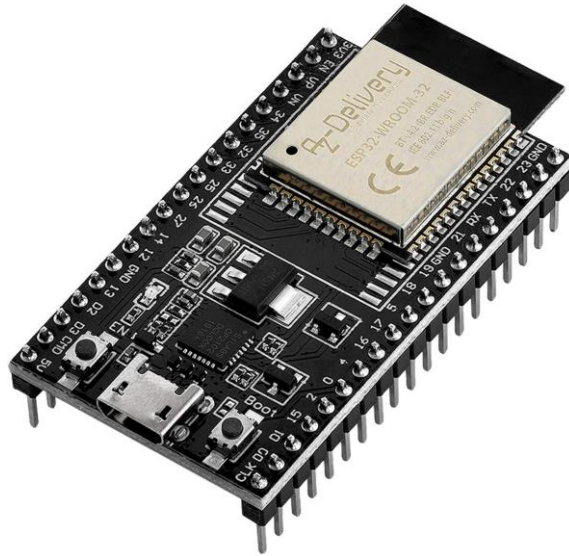


Objetivo 1. Selección de software y hardware



Selección de hardware y software

- Hardware disponible en Guatemala y UVG.

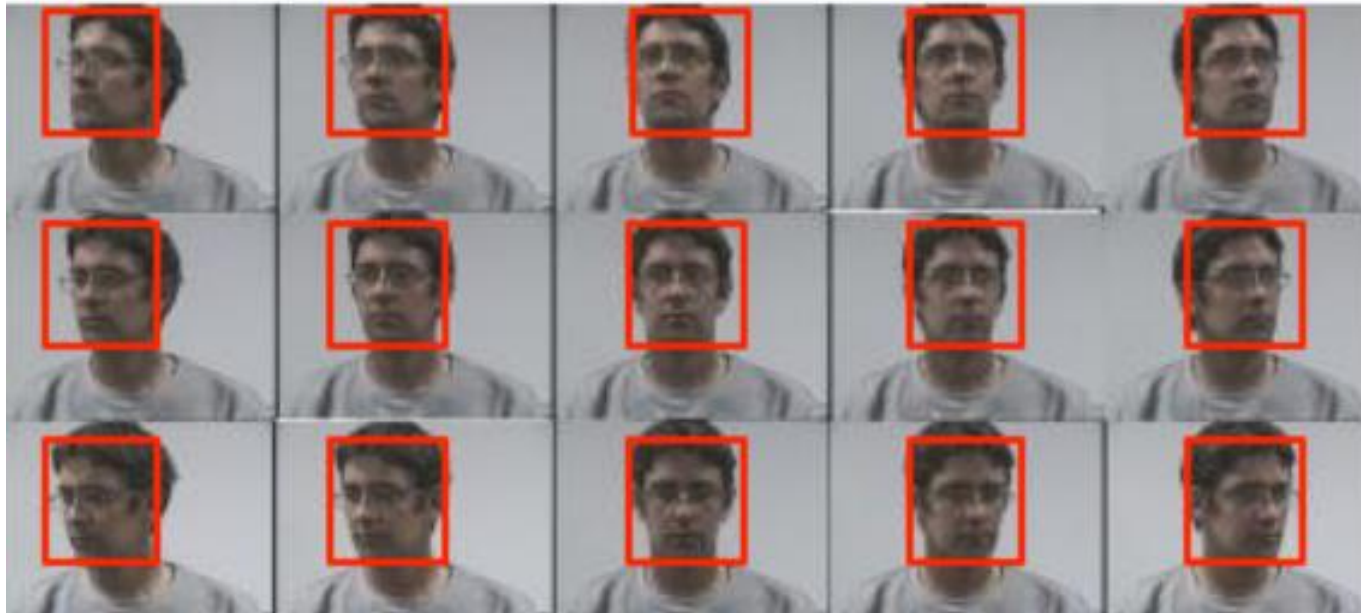


Objetivo 1. Selección de software y hardware



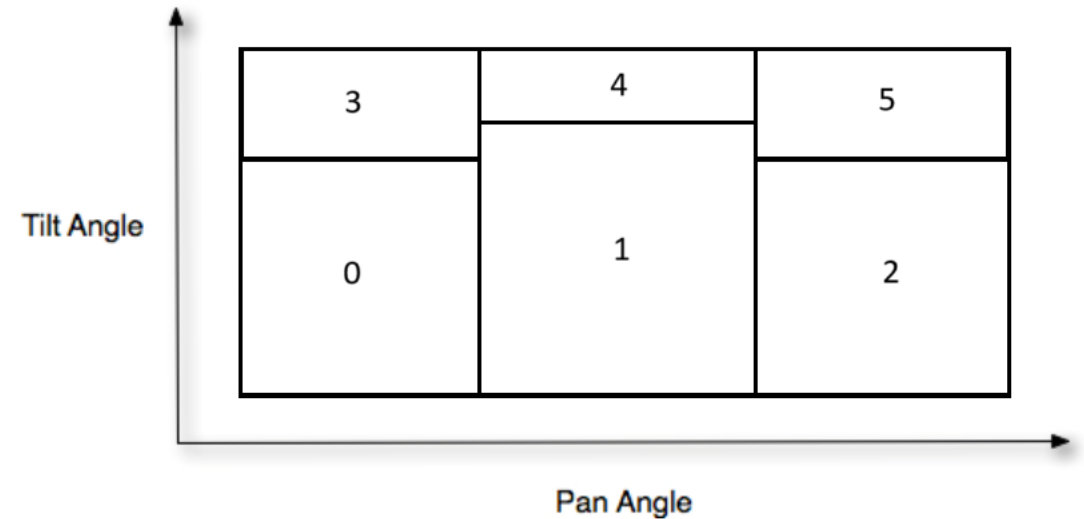
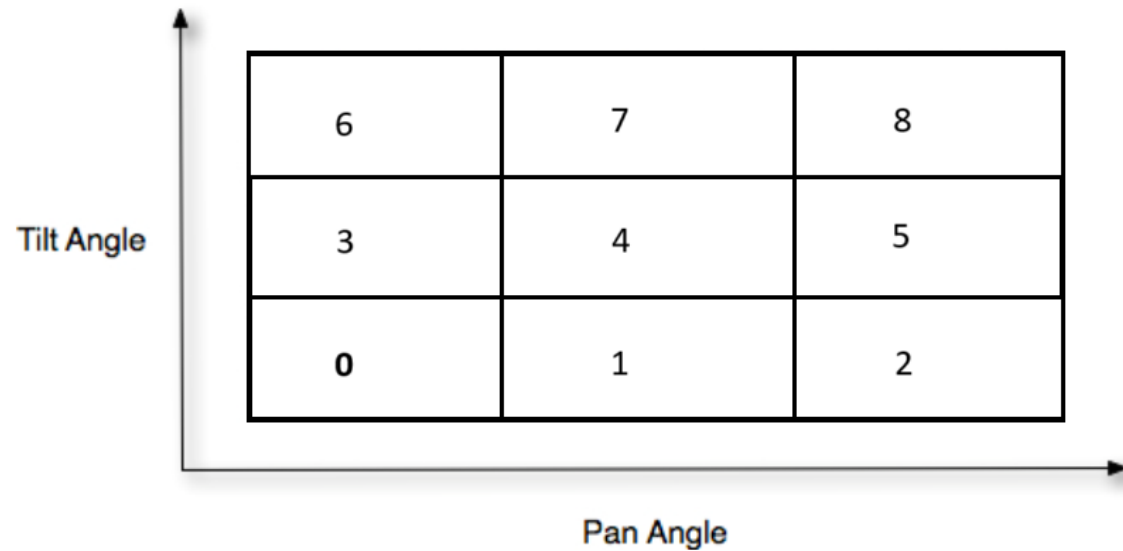
Bases de datos

- ICPR



Bases de datos

- ICPR: Etiquetas



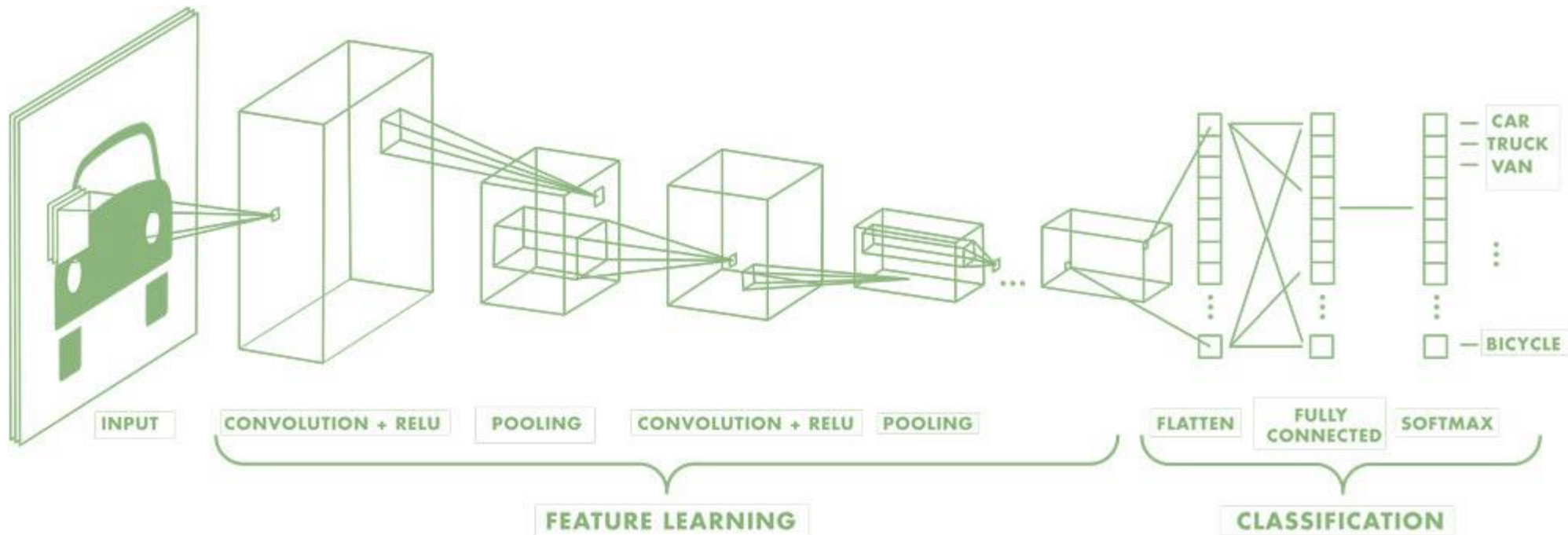
Bases de datos

- Fotografías de compañeros



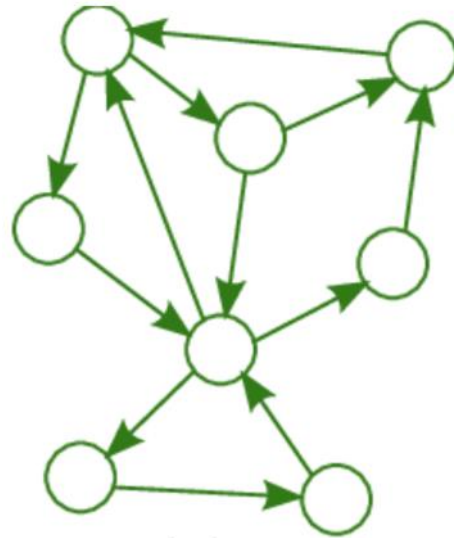
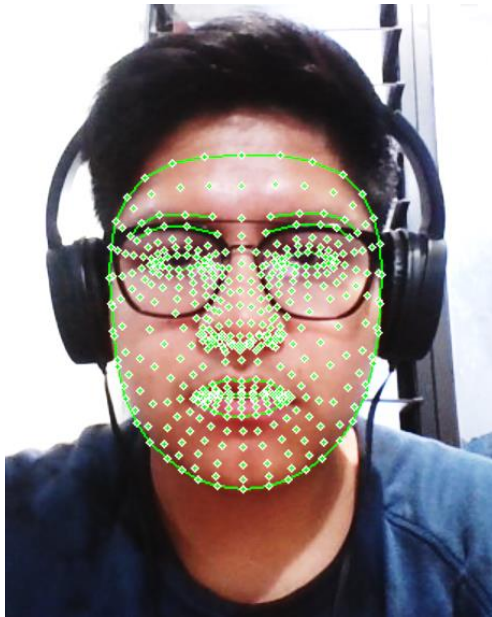
Algoritmos de visión por computadora

- CNN



Algoritmos de visión por computadora

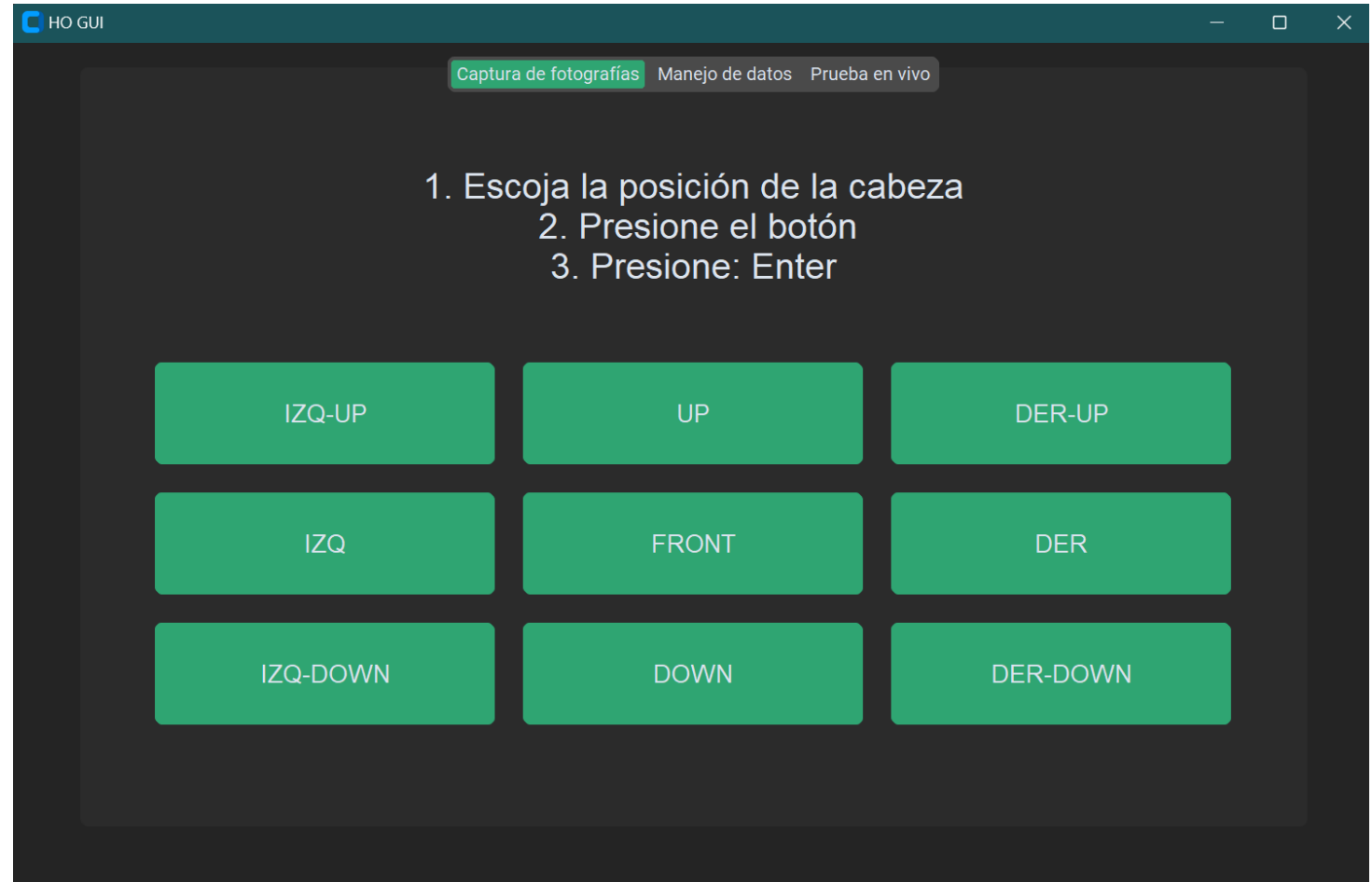
- GNN
- Paro de emergencia



$$c = \sqrt{a^2 + b^2}$$

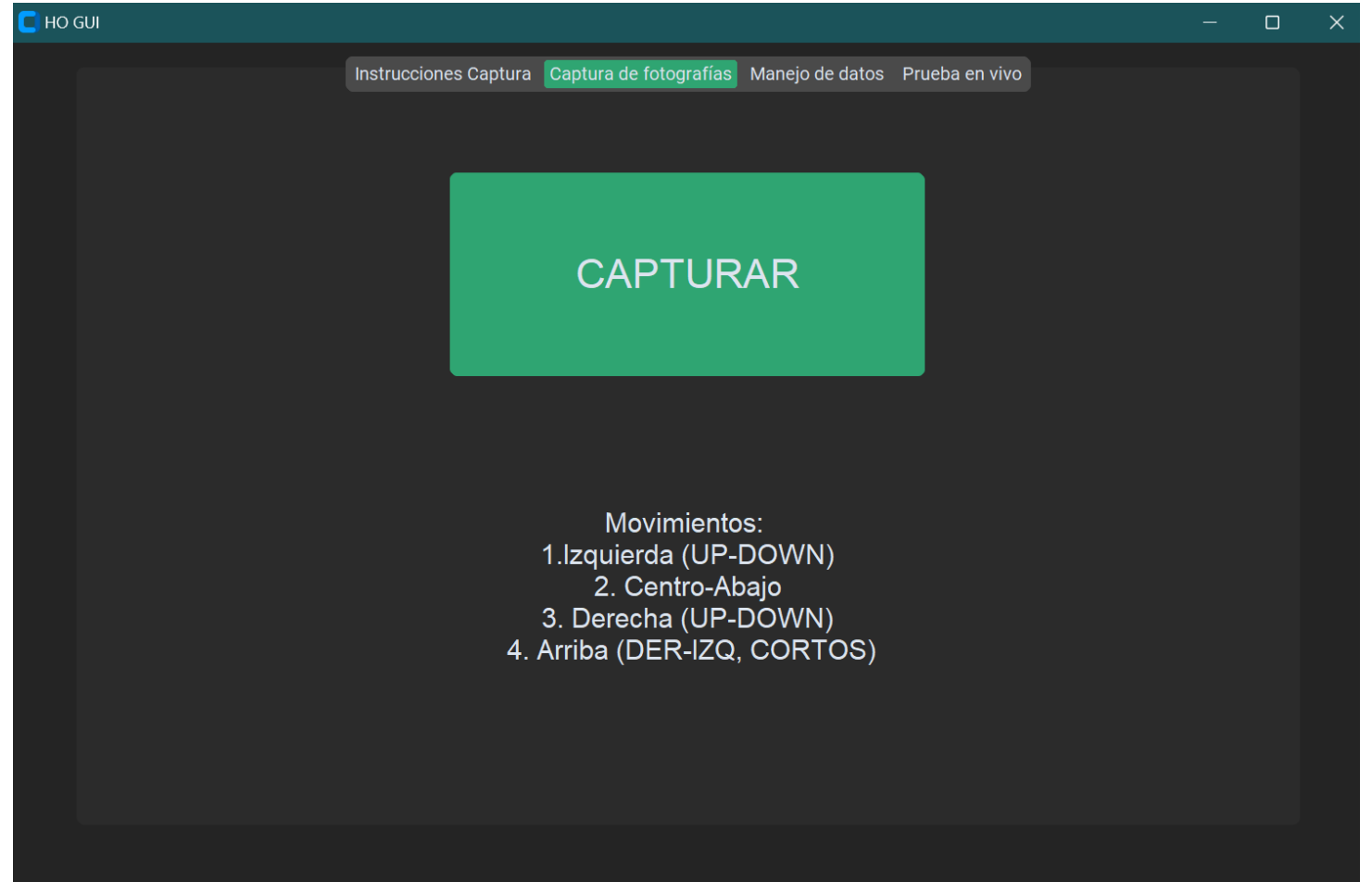
Implementación de interfaz

- Fase 1:
 - Captura de imágenes
 - Transformación de imágenes
 - Transformación de etiquetas
 - Exportación de datos
 - Entrenamiento de aprendizaje de máquina
 - Despliegue de simulación con Turtle
 - Implementación de botones en la interfaz, para cada función



Implementación de interfaz

- Fase 2
 - Mismas funciones que la fase anterior
 - Se añade el filtrado
 - Se optimiza la captura de datos a un solo botón
 - El manejo de datos se limita a 3 botones



Implementación de interfaz

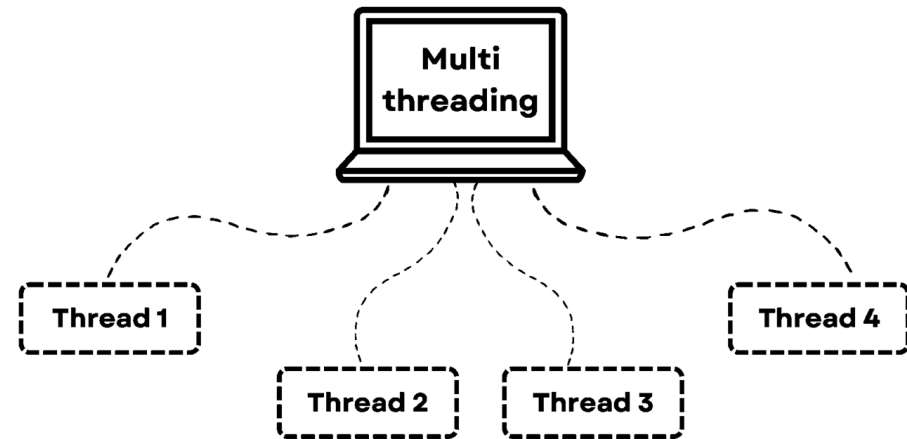
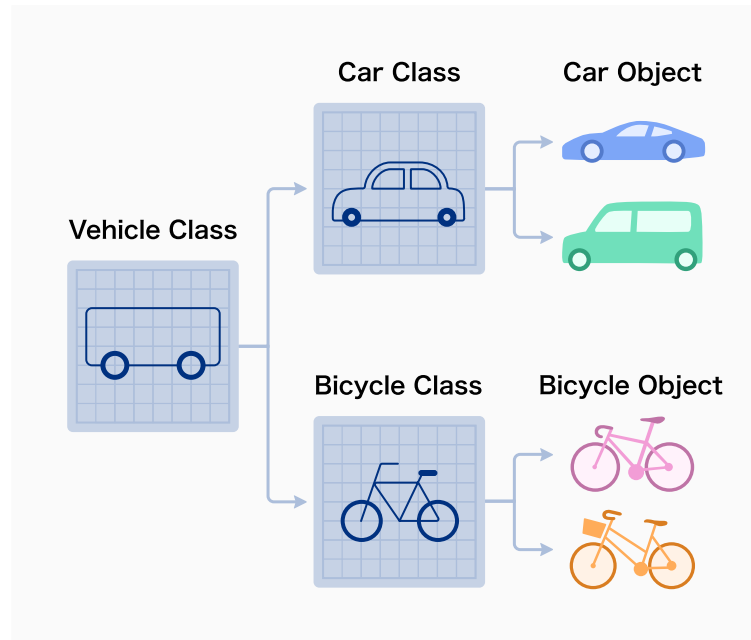
- Fase 3
 - Captura de datos con apoyo audiovisual
 - Integración de botones para prueba con simuladores y ESP32

pciones Captura Captura de fotografías Manejo de datos Prueba Turtle Prueba Webots **Prueba ESP32/POLOLU**

Ejecutar

Detener

OOP y programación multi-hilos



BOARD

Resultados

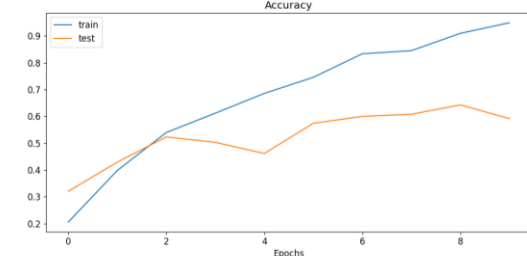
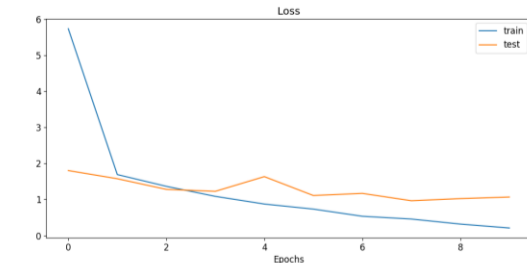
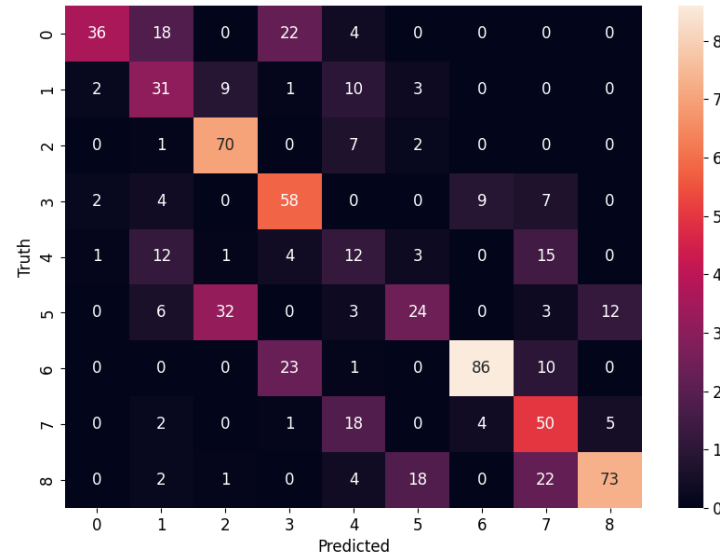
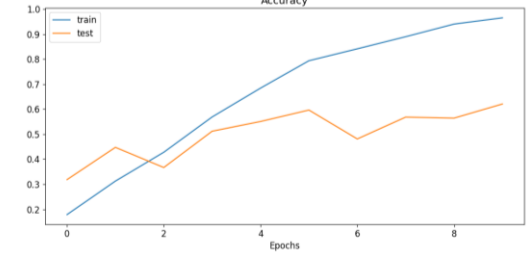
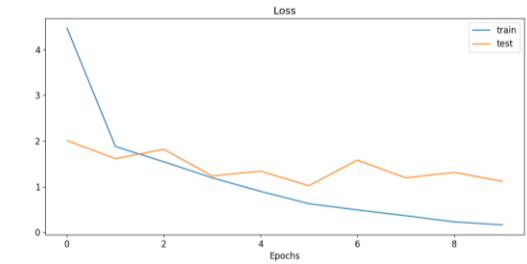
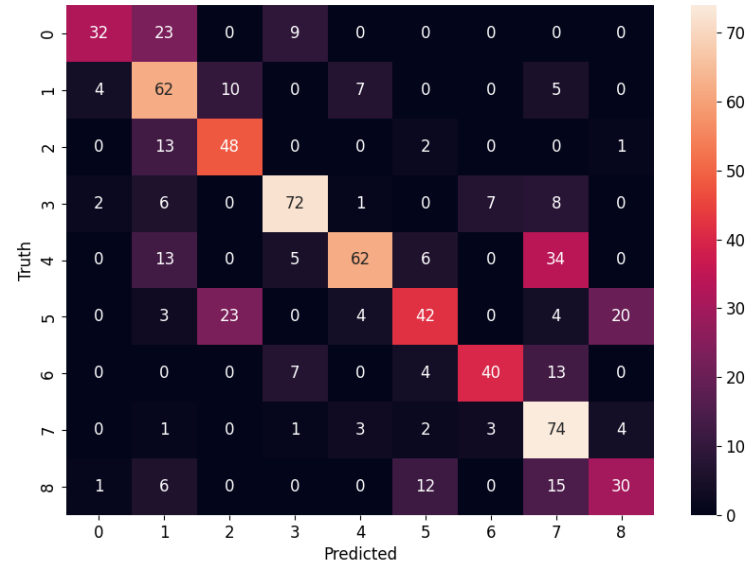


Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

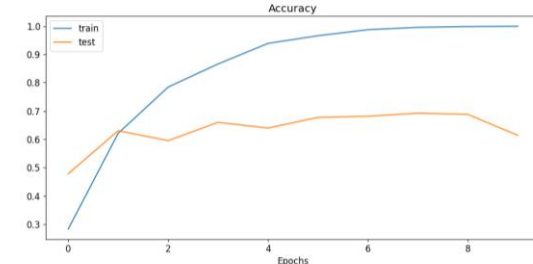
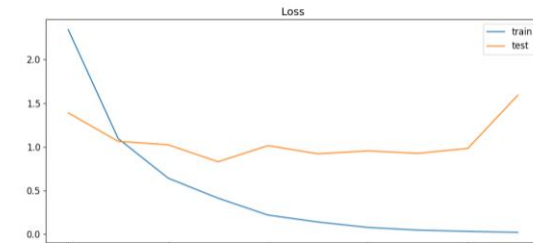
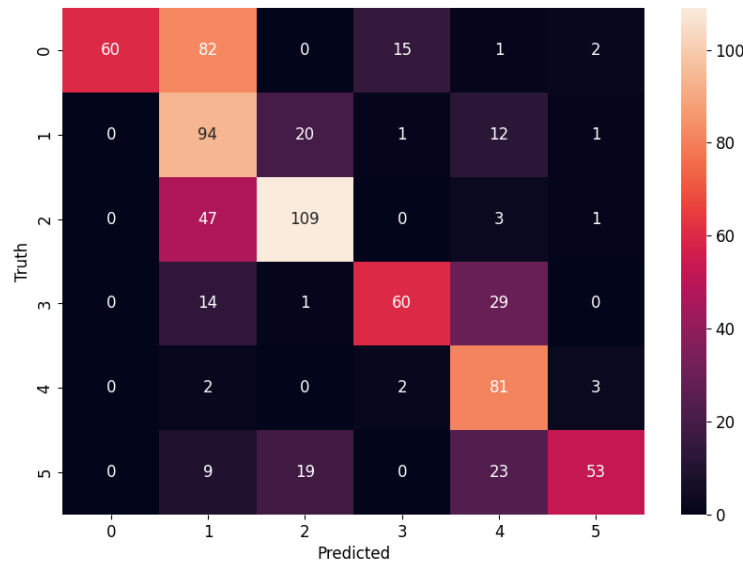
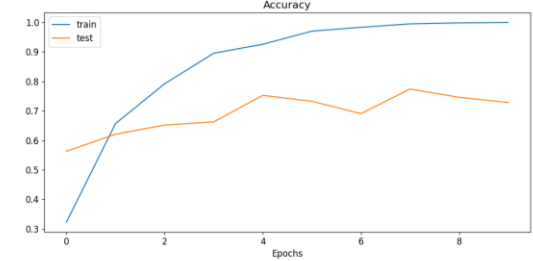
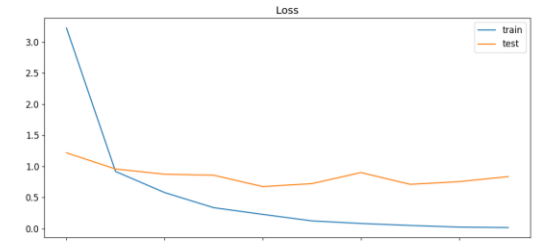
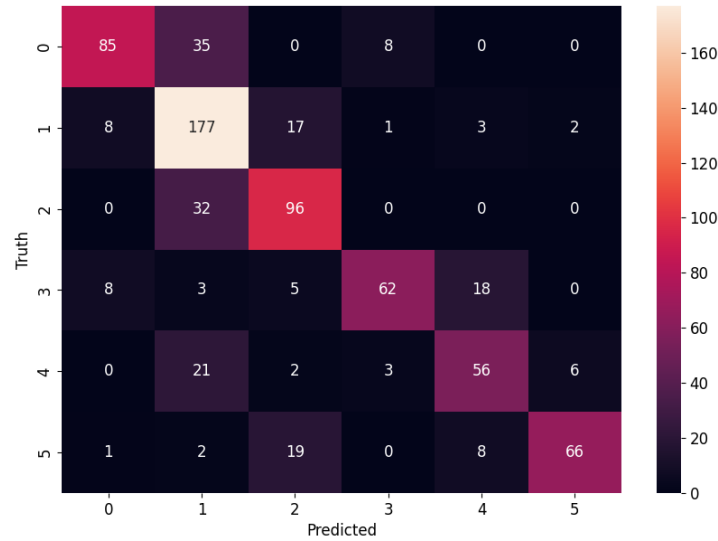
Grupo de modelos 1 (CNN)

- Base de datos:
 - ICPR
- 9 clases (distribución a partir de ICPR)
- Variando distribución de etiquetas



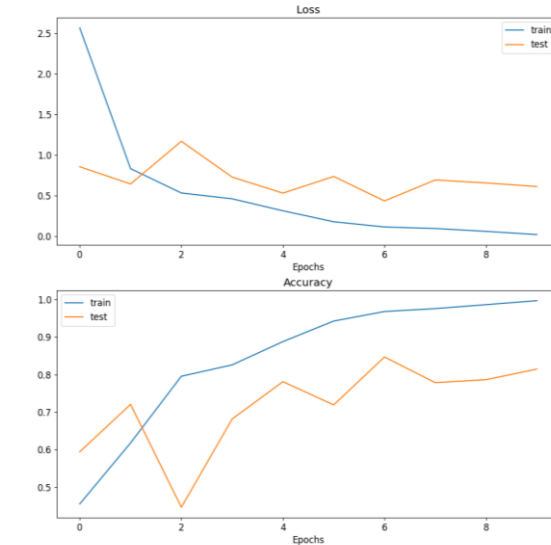
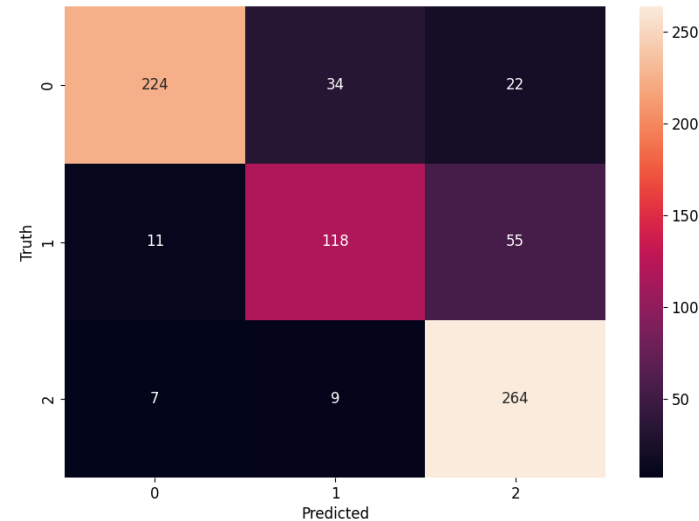
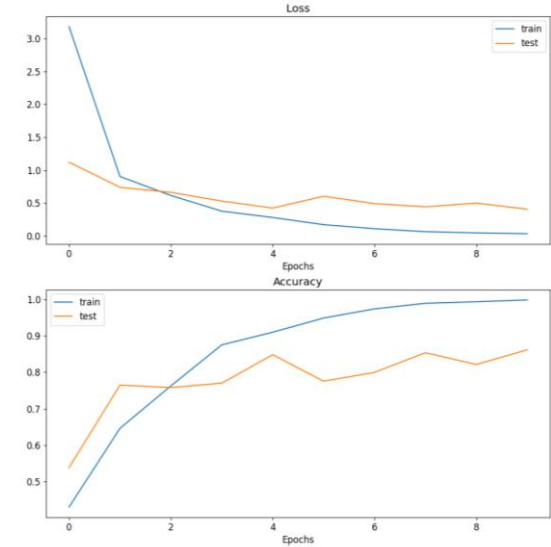
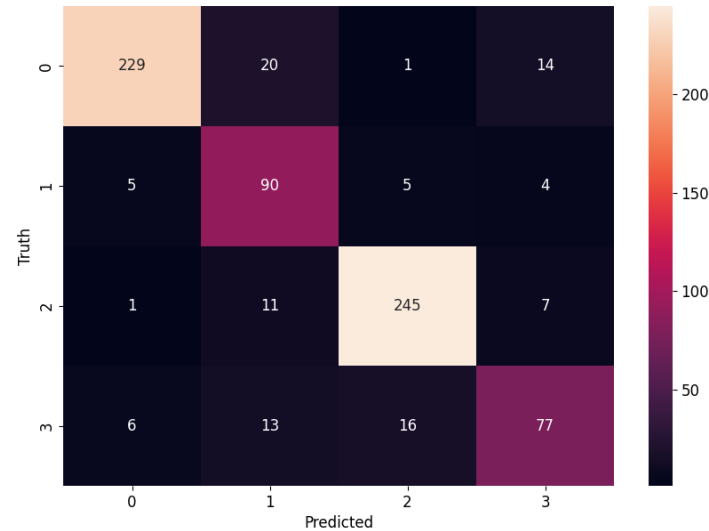
Grupo de modelos 1

- Base de datos:
 - ICPR
- 6 clases (distribución a partir de ICPR)
- Variando distribución de etiquetas



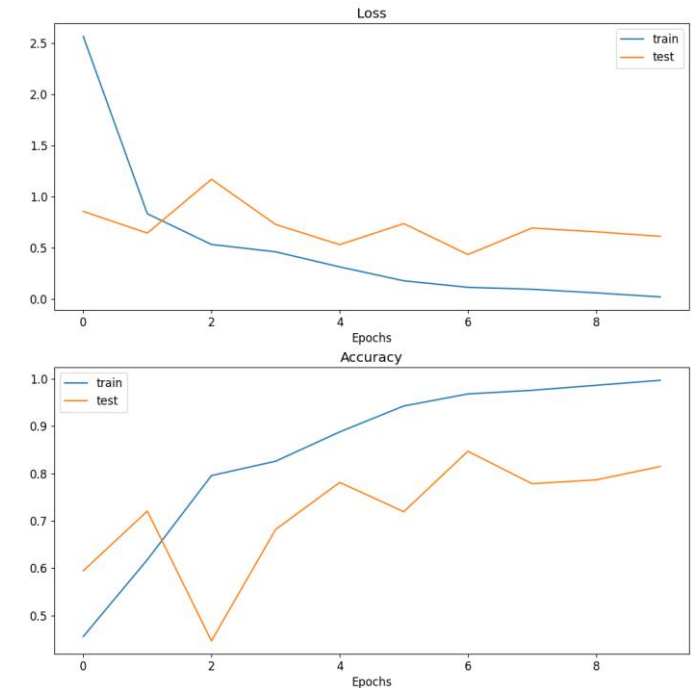
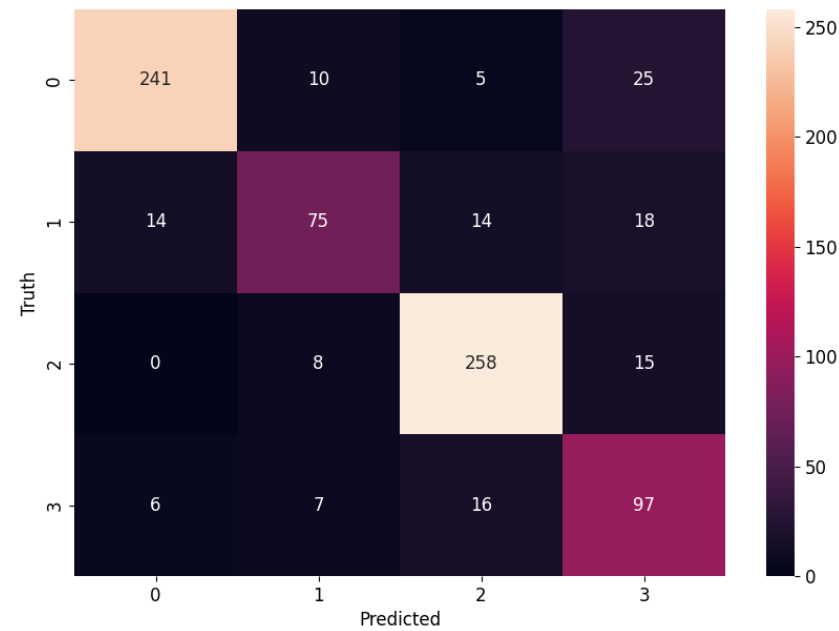
Grupo de modelos 2 (CNN)

- Base de datos:
 - ICPR
- 4 Clases (arriba, centro, derecha, izquierda)
- 3 Clases (centro, derecha, izquierda)



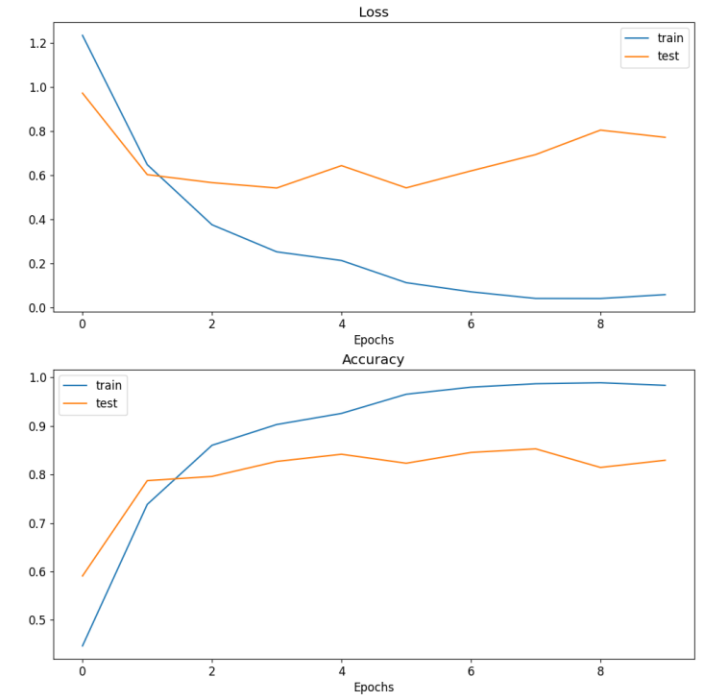
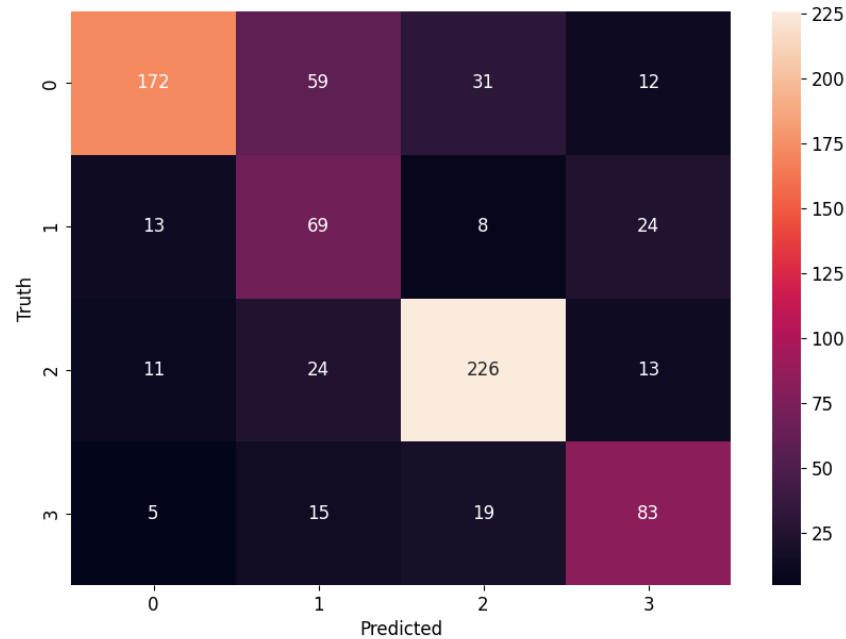
Grupo de modelos 3 (CNN)

- Base de datos:
 - ICPR
 - Fotografías personales
- 4 Clases (arriba, centro, derecha, izquierda)



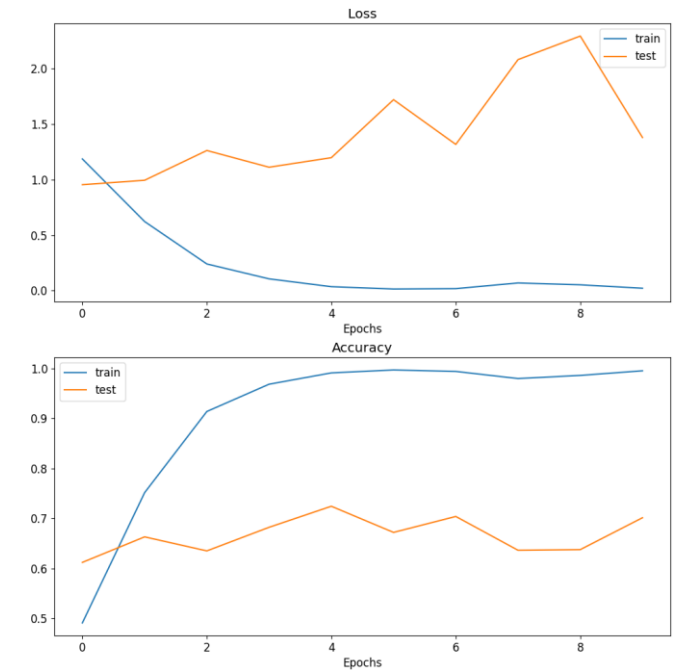
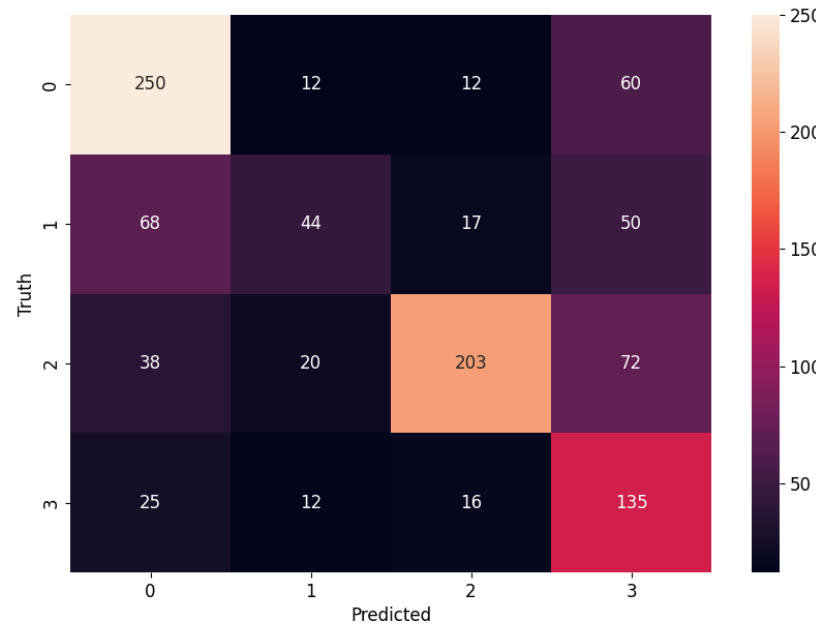
Grupo de modelos 3 (CNN)

- Base de datos:
 - ICPR
 - Fotografías personales con filtros
- 4 Clases (arriba, centro, derecha, izquierda)



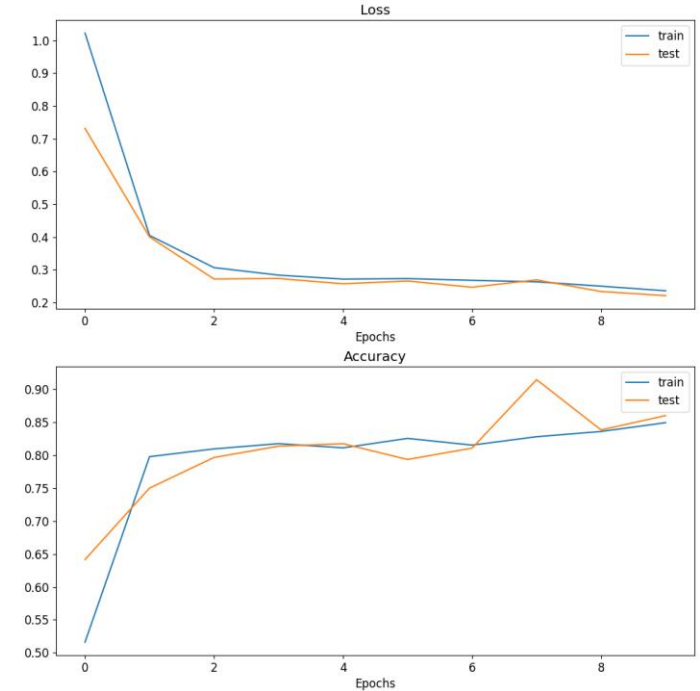
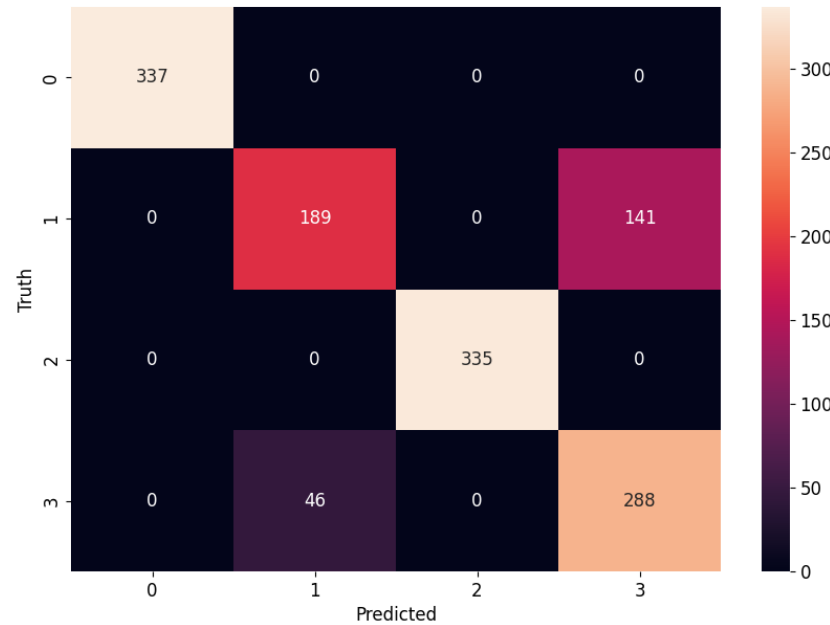
Grupo de modelos 3 (CNN)

- Base de datos:
 - ICPR
 - Fotografías personales con marcadores físicos
- 4 Clases (arriba, centro, derecha, izquierda)



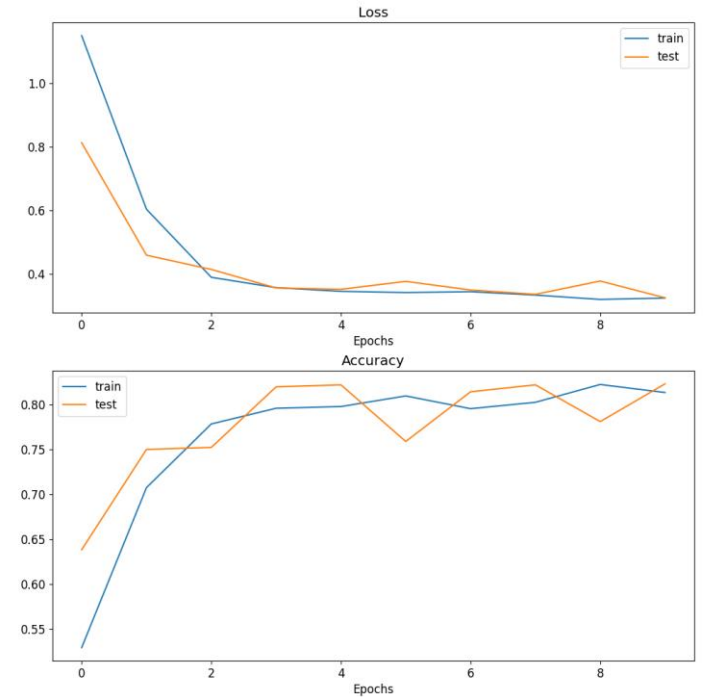
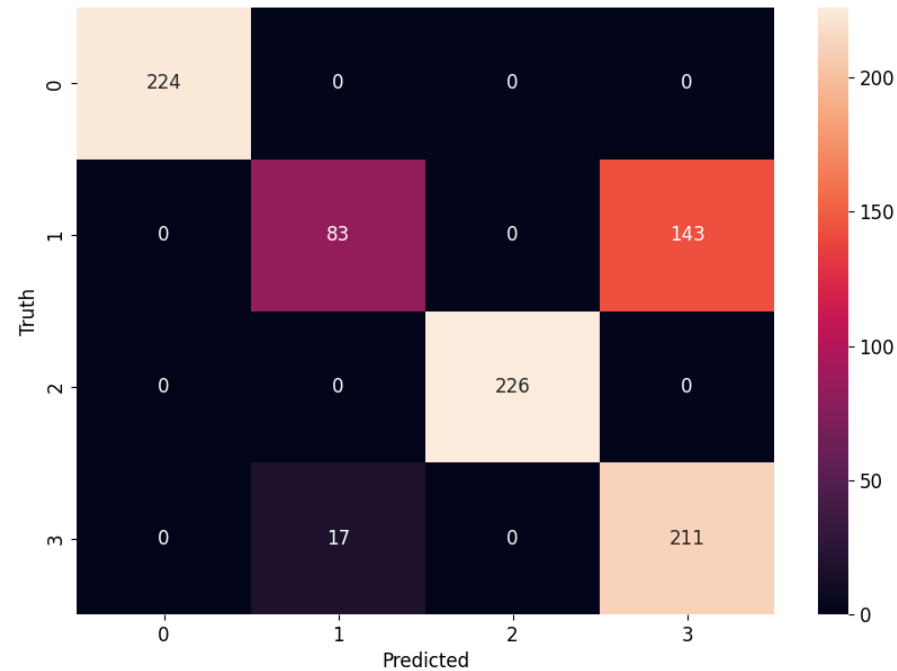
Grupo de modelos 4 (GNN)

- Base de datos:
 - Fotografías personales
- 4 Clases (centro, derecha, izquierda, sonrisa)



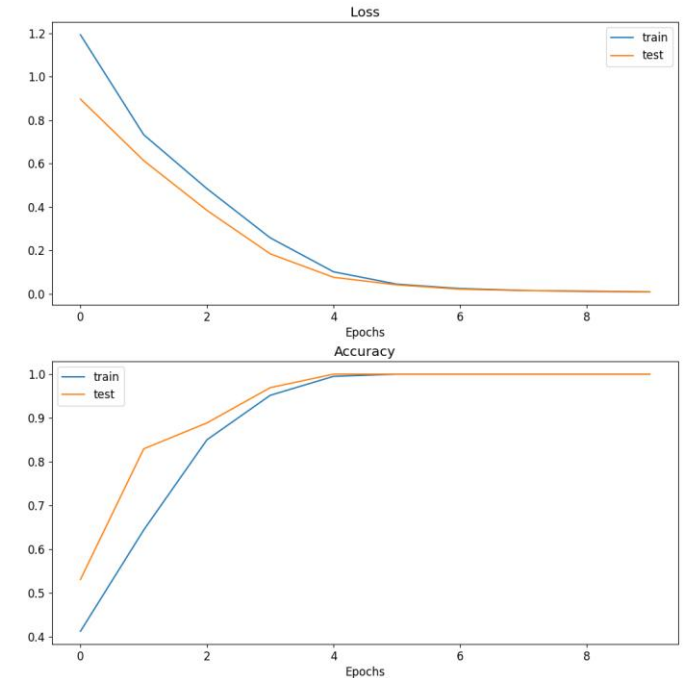
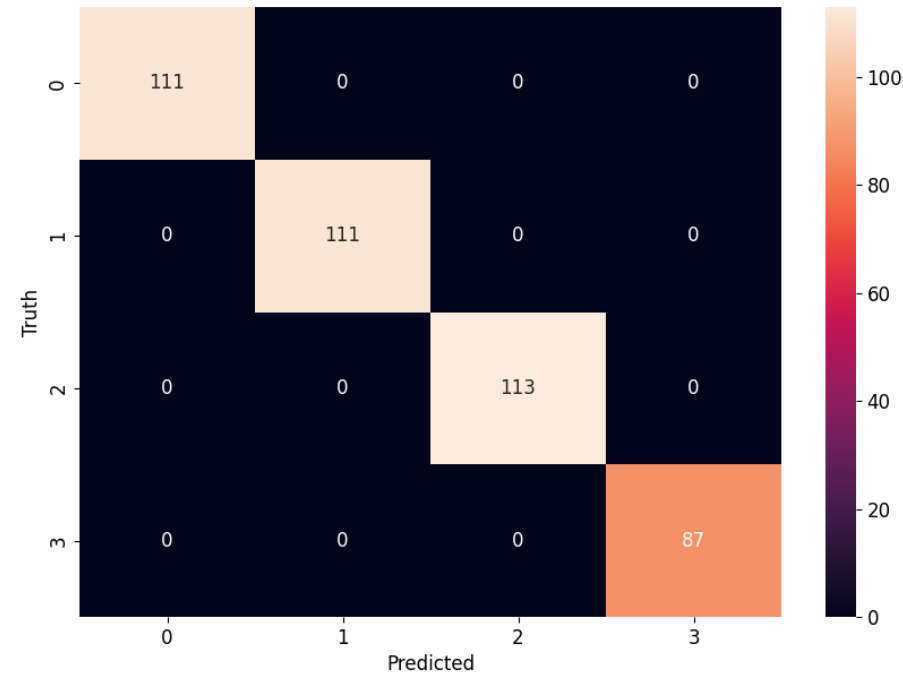
Grupo de modelos 4 (GNN)

- Base de datos:
 - Fotografías personales
- 4 Clases (centro, derecha, izquierda, boca abierta)



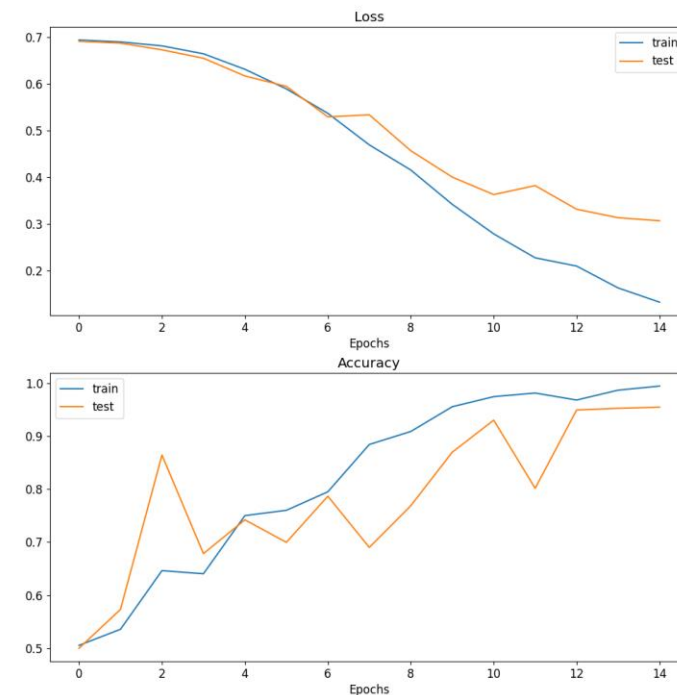
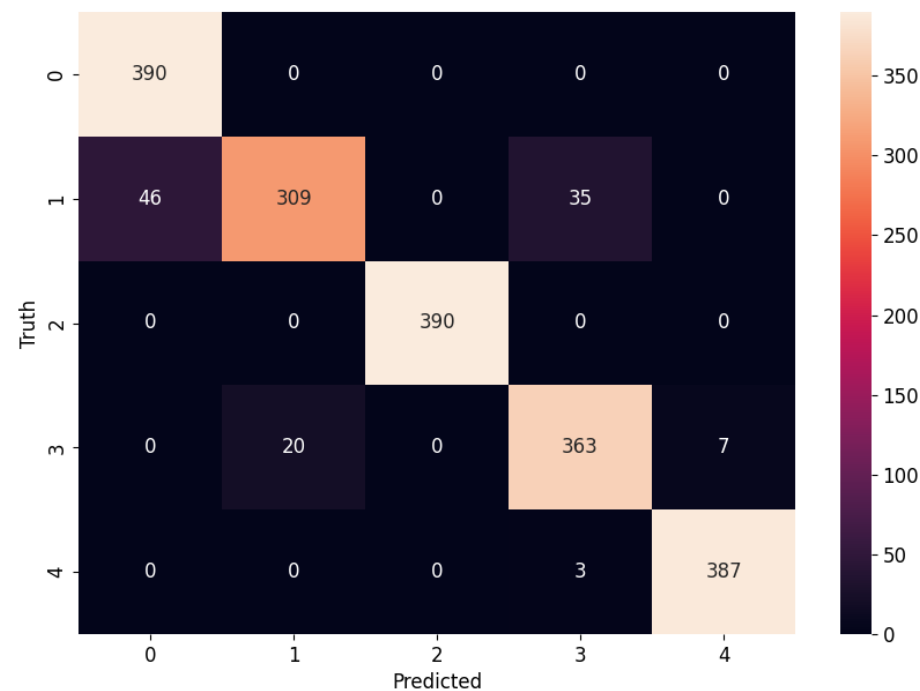
Grupo de modelos 4 (GNN)

- Base de datos:
 - Fotografías personales
- 4 Clases (arriba, centro, derecha, izquierda)



Grupo de modelos 4 (GNN)

- Base de datos:
 - Fotografías personales
- 5 Clases (arriba, centro, derecha, izquierda, abajo)

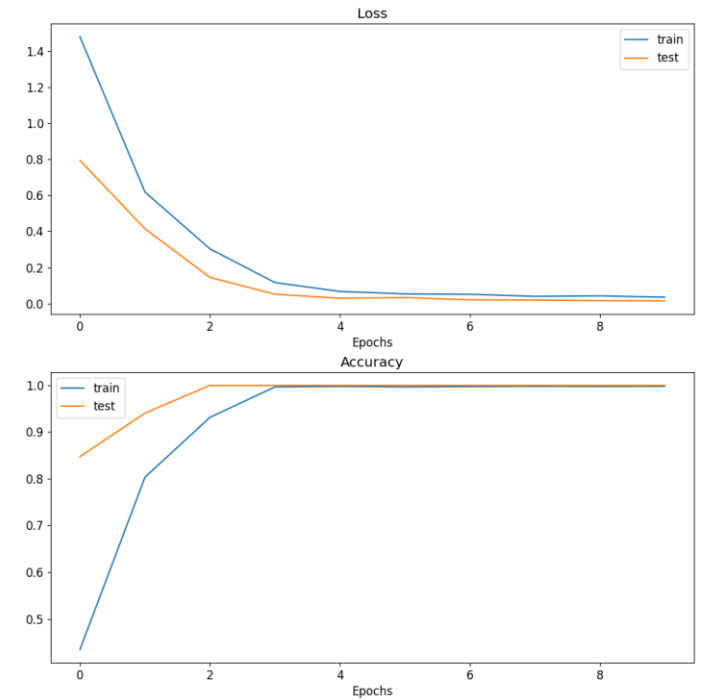
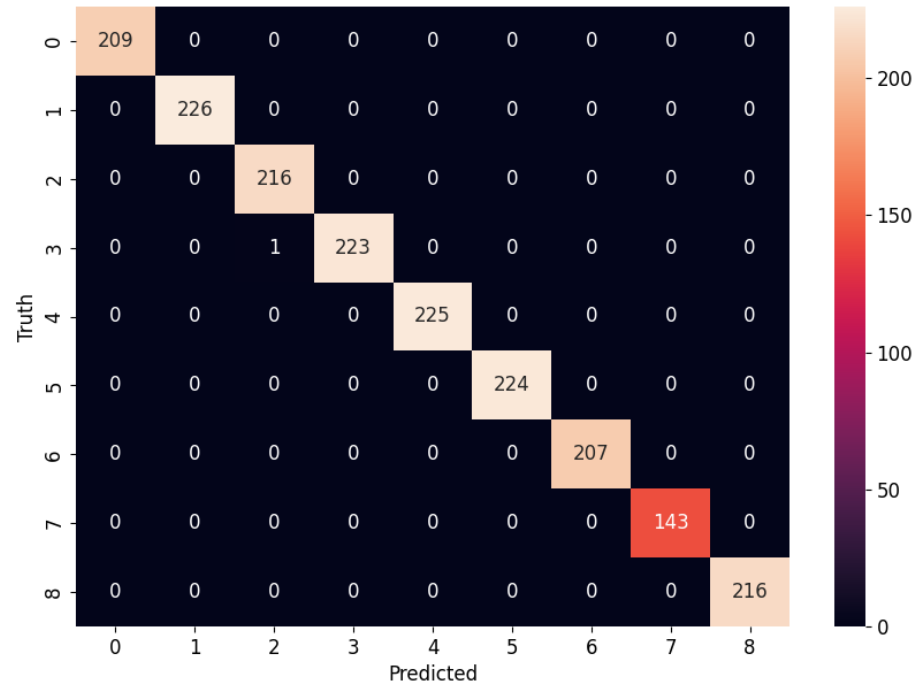


Objetivo 2. Algoritmo de reconocimiento de gestos por visión de computadora



Grupo de modelos 4 (GNN)

- Base de datos:
 - Fotografías personales
- 9 Clases
(distribución a partir de ICPR)

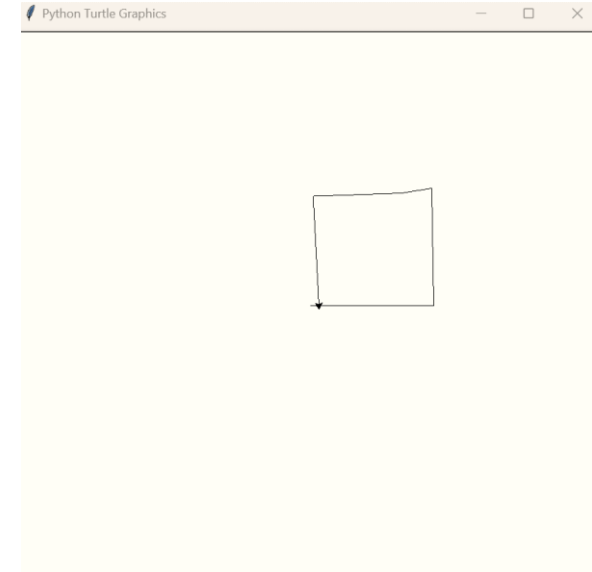


Simuladores

- Turtle:
antes de
grupo 4



- Turtle:
después de
grupo 4



Objetivo 3. Algoritmo de
traducción de gestos

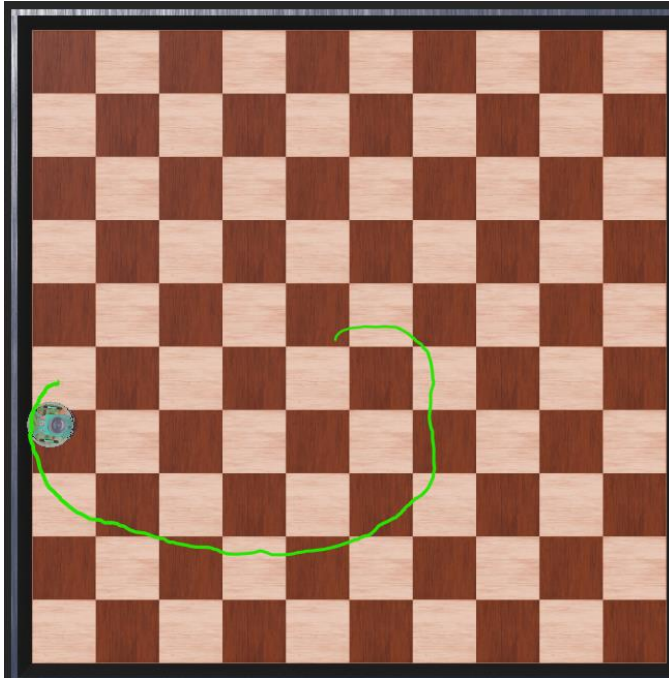


Objetivo 4. Validación en
simuladores



Simuladores

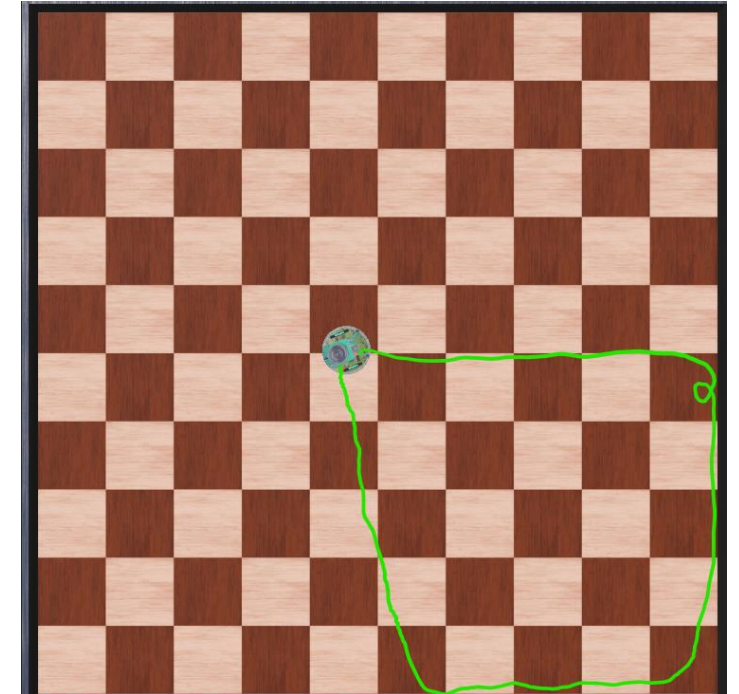
- Webots:
antes de
grupo 4



Objetivo 3. Algoritmo de traducción de gestos



- Webots:
después de
grupo 4

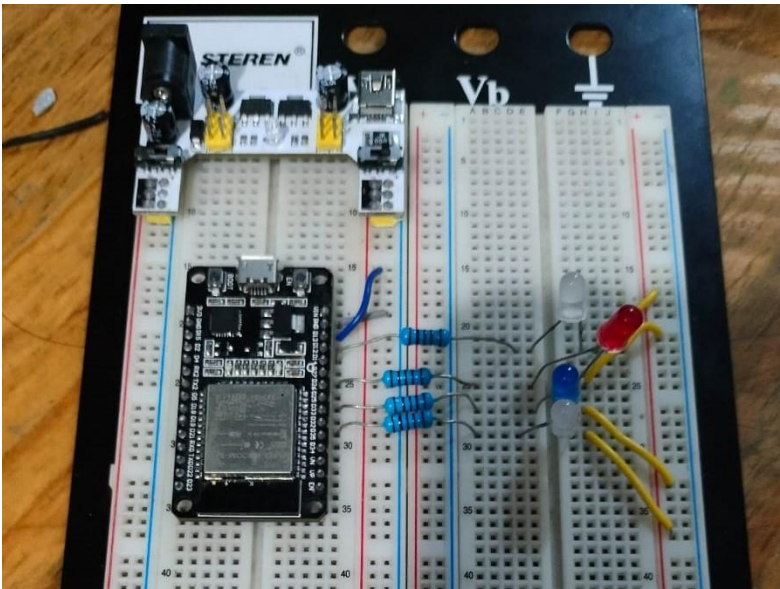


Objetivo 4. Validación en simuladores



Plataforma física

- ESP32 DevKit C v4



Objetivo 3. Algoritmo de traducción de gestos

- ESP/POLOLU 3PI+



Objetivo 5. Validación en plataforma robótica móvil



Demostraciones

- Captura de datos:
<https://www.youtube.com/watch?v=zi-wfVPRmNw>
- Plataforma robótica móvil P1:
<https://www.youtube.com/watch?v=BH5b8ZK79zQ>
- Plataforma robótica móvil P2:
<https://www.youtube.com/watch?v=UJ2htHZI7T8>

Conclusiones y Recomendaciones



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Conclusiones

- La selección de hardware resultó útil, dado que todas las pruebas fueron realizadas con la cámara de la laptop, y la verificación física únicamente requirió una Protoboard, cables, diodos emisores de luz, una fuente de alimentación y la placa ESP32.
- La selección del software fue útil durante el proyecto, dado que se pudo experimentar con diferentes tipos de librerías para completar las funciones requeridas por el sistema y plataformas como Mediapipe prometen seguir en el mercado por varios años.
- Se validó que los algoritmos de visión por computadora no tuvieron resultados exitosos cuando se combinaron métodos de reconocimiento clásicos, como filtros o marcadores físicos en las imágenes de las bases de datos para alimentar los modelos de aprendizaje de máquina.

Conclusiones

- El algoritmo de visión de computadora tuvo resultados exitosos gracias a la combinación de la herramienta Mediapipe, la cual permitió obtener los puntos de la cara, la cual se puede interpretar como un grafo y la implementación de la red neuronal de grafos.
- El algoritmo de traducción de gestos fue creado pensando en el movimiento de un automóvil de velocidades, lo cual funcionó de manera satisfactoria, al utilizar los gestos de movimiento de cara hacia arriba y abajo, así como controlar la dirección con el movimiento hacia izquierda o derecha.
- La validación por medio de simulaciones pudo implementarse de manera correcta al programar el graficador Turtle y al configurar de manera exitosa la conexión entre Webots y el entorno de programación PyCharm Community Edition como un controlador externo.

Conclusiones

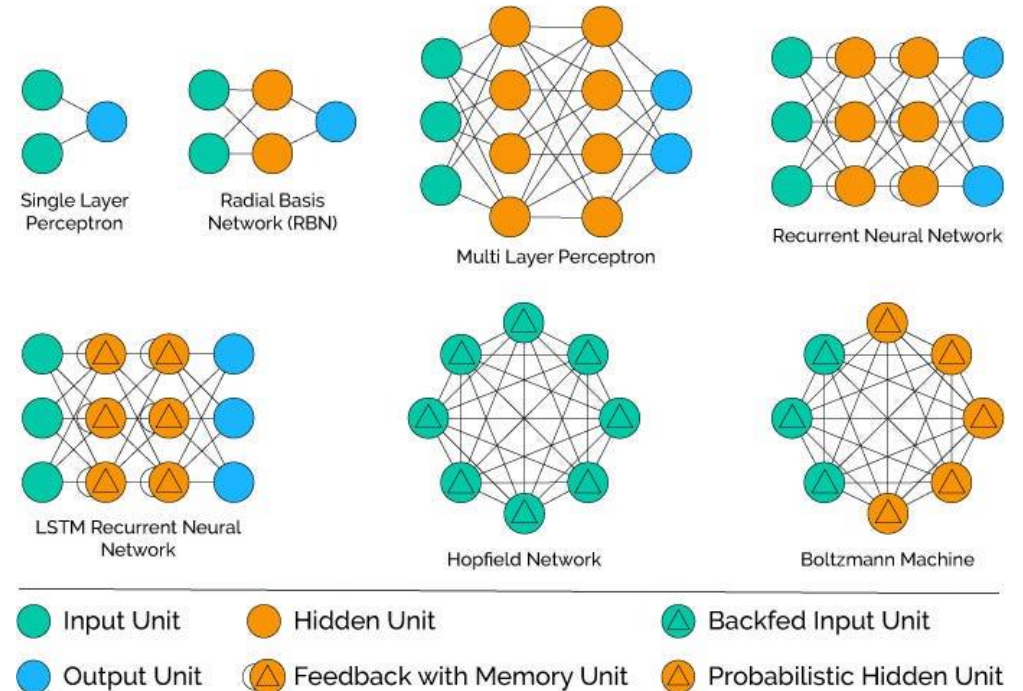
- La validación por medio de comunicación Bluetooth, permitió determinar que el envío de datos fuera correcto utilizando la cámara de la laptop y una placa ESP32 que se encarga de manejar los comandos a utilizar en la plataforma robótica.
- El costo computacional de utilizar el lenguaje de programación Python, se ve reflejado cuando no se programa adecuadamente el hilado de procesos, ocasionando congelamiento de funciones.
- El costo computacional de trabajar con Python se ve compensado con la forma sencilla e intuitiva para configurar redes neuronales de diferentes tipos como convolucionales y de grafos.

Conclusiones

- Las simulaciones realizadas desde la plataforma Webots, son valiosas, sin embargo, para un modelo predictivo la simulación con Turtle genera una representación útil para la verificar el funcionamiento del modelo.
- Usar programación orientada a objetos permitió implementar una interfaz gráfica la cual es fácilmente depurable.

Recomendaciones

- Migración a otros lenguajes de programación más eficientes, por ejemplo C.
- Integración al Optitrack para aplicaciones como seguimiento de posición.
- Para obtener un mejor registro de las observaciones de las pruebas realizadas, se recomienda crear una bitácora con registro detallado de cada prueba con algoritmos.



Recomendaciones

- Exploración de diferentes modelos de aprendizaje, o diferentes enfoques, dado que se trabajó con dos predictivos.
- Optimización de la interfaz, así como su exportación a aplicación para PC o dispositivo móvil.
- Explotar el potencial explorativo utilizando otro tipo de comunicaciones, para implementación en IoT.
- Implementar un sistema de seguridad automático que evite colisiones con obstáculos.



Referencias Bibliográficas

- [1] E. Pichon Riviére, “Patogenia y dinamismos de la epilepsia,” 2017
 - [2] S.-W. Lee, Automatic Gesture Recognition for Intelligent Human-Robot Interaction. Mayo de 2006. dirección: <https://ieeexplore.ieee.org/document/1613091>
 - [3] S. Sharma, S. Jain y Khushboo, A Static Hand Gesture and Face Recognition System for Blind People, en 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, págs. 534-539. doi: 10.1109/SPIN.2019.8711706.
 - [4] R. T. Bankar y S. S. Salankar, Head Gesture Recognition System Using Gesture Cam, en 2015 Fifth International Conference on Communication Systems and Network Technologies, 2015, págs. 535-538. doi: 10.1109/CSNT.2015.81.
 - [5] L. A. Rivera, G. N. DeSouza y L. D. Franklin, Control of a Wheelchair Using an Adaptive K-Means Clustering of Head Poses, en 2013 IEEE Symposium on Computational Intelligence in Rehabilitation and Assistive Technologies (CIRAT), 2013, págs. 24-31. doi: 10.1109/CIRAT.2013.6613819.
-

Referencias Bibliográficas

[6] Dirección: <https://developers.google.com/mediapipe>

[7] Aggarwal, Neural networks and deep learning. dirección: <https://link.springer.com/book/10.1007/978-3-319-94463-0>

[8] IBM, What is Computer Vision? n.d. dirección: <https://www.ibm.com/topics/computer-vision>

[9] A. S. Huang y L. Rudolph, Bluetooth essentials for programmers. Cambridge University Press, 2007.

[10] L. J. Zhi, Sistemas operativos en tiempo real (RTOS) y Sus Aplicaciones, feb. de 2021. dirección: <https://www.digikey.com/es/articles/real-time-operating-systems-and-their-applications>

¡Gracias por su atención!

(Inicia espacio para preguntas de la Terna)