

SIS330

Desarrollo de Aplicaciones Inteligentes

**"BuddyBot, Conducción Autónoma y
Compañía Inteligente"**

PRESENTA

Marcani Aguilar Abel Israel

TUTOR

Ing. Pacheco Lora Carlos

Sucre – Bolivia

2024

Índice

1. Resumen.....	3
2. Antecedentes.....	3
3. Problema principal.....	3
4. Objetivo general.....	3
5. Objetivos Específicos.....	3
6. Abordaje de la solución.....	4
7. Fundamentos teóricos considerados en el trabajo.....	4
7.1. Ámbito de la inteligencia artificial.....	4
7.2. Ámbito al que se aplicó la inteligencia artificial.....	5
8. Metodología.....	5
8.1. Sustento metodológico.....	5
8.2. Técnicas de recolección de datos.....	6
8.3. Materiales & herramientas.....	6
8.3.1. Arquitectura del proyecto desarrollado.....	6
8.3.2. Esquema y descripción de componentes de hardware.....	7
8.3.3. Esquema y descripción de modelos o componentes inteligentes.....	8
8.3.4. Valores de parámetros e hiperparámetros aplicados.....	9
8.3.5. Especificaciones técnicas.....	10
8.3.6. Lenguajes de programación, frameworks, entre otros.....	10
8.4. Plan de trabajo.....	11
9. Resultados.....	11
9.1. Dataset.....	11
9.1.1. Descripción y preprocesamiento realizado.....	11
9.1.2. Conjunto de entrenamiento, evaluación y validación.....	13
9.1.3. Técnicas, criterio y/o métodos aplicados para la conformación de los conjuntos de datos de entrenamiento, prueba y validación.....	13
9.2. Resultados de entrenamiento y prueba (Métricas de rendimiento).....	14

9.2.1.	Matriz de confusión	14
9.2.2.	Perdidas y Presición (UNet)	14
9.2.3.	Curva de aprenizaje (PPO)	15
9.3.	Resultados de aplicación	15
10.	Conclusiones	16
11.	Recomendaciones	16
12.	Referencias bibliográficas.....	17



1. Resumen

Este proyecto aborda el creciente problema del sedentarismo y desconexión social derivado de la dependencia de las pantallas. La solución propuesta consiste en BuddyBot, un robot interactivo equipado con conducción autónoma. BuddyBot se presenta como una alternativa innovadora que busca contrarrestar los efectos negativos de la era digital, fomentando la actividad física y promoviendo interacciones sociales significativas. El enfoque técnico se basa en algoritmos de aprendizaje por refuerzo, redes neuronales convolucionales y la plataforma Jetson Nano de NVIDIA.

2. Antecedentes

En la actualidad, la sociedad enfrenta un preocupante aumento del sedentarismo y la adicción a las pantallas, especialmente entre niños y jóvenes, con consecuencias negativas para la salud física y mental, así como para el desarrollo social. Estudios significativos, como "Influencia de la tecnología en la interacción social adolescente" (2020) y "Impacto del sedentarismo en la salud mental" (2022), revelan que el uso excesivo de pantallas se vincula con aislamiento social, ansiedad y depresión en los adolescentes, destacando la importancia crucial de la interacción cara a cara en el desarrollo de la identidad y habilidades sociales. Además, investigaciones como "El impacto de las redes sociales en la salud mental de los jóvenes" (2021) subrayan el riesgo de soledad y baja autoestima asociado al uso intensivo de redes sociales. Por otro lado, "La salud mental en la era digital" (2022) destaca los beneficios terapéuticos de la interacción social cara a cara, reduciendo el estrés, la ansiedad y la depresión, y mejorando el bienestar general. Estos estudios subrayan la urgencia de soluciones que promuevan una vida más activa y conectada, abordando los desafíos contemporáneos de la dependencia digital y el sedentarismo.

3. Problema principal

Debido a la dependencia de las pantallas, nos enfrentamos a un problema creciente de sedentarismo y desconexión social. La falta de interacción real tiene un impacto negativo en la salud física y mental de las personas. Frente a esta realidad, BuddyBot es una nueva respuesta para este problema y ofrece una experiencia única que ayuda a contrarrestar los efectos negativos de la era digital en nuestras vidas.

4. Objetivo general

Mejorar significativamente la calidad de vida de los usuarios a través de BuddyBot, un robot interactivo con capacidades de conducción autónoma y guiado personalizado. Proporcionar experiencias que no solo satisfagan las necesidades prácticas de los usuarios, sino que también generen momentos de alegría y distracción, contribuyendo así al bienestar y la felicidad en su vida cotidiana.

5. Objetivos Específicos

- Establecer los fundamentos teóricos necesarios en el ámbito de la robótica y la inteligencia artificial, proporcionando un marco teórico sólido para la implementación de mecanismos de control en BuddyBot que permitan un desempeño óptimo y adaptado a su contexto de conducción autónoma.

- Investigar e implementar modelos avanzados de aprendizaje profundo, como Proximal Policy Optimization (PPO) y Convolutional Neural Networks (CNN), para mejorar la conducción autónoma y la experiencia de usuario.
- Desarrollar un prototipo funcional de BuddyBot que incorpore algoritmos de aprendizaje por refuerzo para la detección de obstáculos y la navegación autónoma.
- Realizar pruebas exhaustivas para validar la efectividad de los modelos de inteligencia artificial en BuddyBot, ajustándolos según los resultados. El objetivo es garantizar un rendimiento óptimo tanto en conducción autónoma.

6. Abordaje de la solución

El algoritmo de aprendizaje por refuerzo PPO (Proximal Policy Optimization) será la base de la conducción autónoma de BuddyBot. La similitud del entorno doméstico con un escenario complejo y dinámico, adecuado para aprovechar las capacidades de aprendizaje por refuerzo, respalda esta decisión. Las pruebas comparativas demostrarán la eficacia y adaptabilidad de este método. Estas pruebas evaluarán la velocidad y eficacia del entrenamiento en comparación con otras opciones, así como su capacidad para generalizar el aprendizaje a diferentes situaciones.

Una red neuronal convolucional (CNN) se implementará para procesar información visual. Las pruebas de análisis de características del entorno capturadas por la cámara de BuddyBot verificarán su funcionamiento. Durante las fases de entrenamiento y pruebas, se investigarán opciones adicionales como TRPO (Optimización de Políticas por Trust Region) y DDPG (Deep Deterministic Policy Gradients) además del PPO para evaluar su desempeño y determinar la mejor opción basada en eficacia y eficiencia.

La plataforma de hardware Jetson Nano de NVIDIA fue seleccionada por su capacidad para procesar información visual de la CNN y ejecutar algoritmos de aprendizaje por refuerzo. Esta validación técnica completa asegurará que BuddyBot sea autónomo y adaptable a su entorno, brindándole una experiencia de conducción autónoma avanzada.

Para el procesamiento de modelos avanzados, utilizaremos un servidor en el que se alojarán todos los modelos necesarios. BuddyBot se conectará a este servidor para consumir sus servicios. Esta decisión se tomó debido a que los modelos que emplearemos son demasiado grandes para ser soportados por la Jetson Nano, especialmente durante la etapa de entrenamiento. En cuanto a la inferencia, realizaremos pruebas exhaustivas para evaluar y optimizar los resultados.

7. Fundamentos teóricos considerados en el trabajo

Los fundamentos teóricos del campo de la inteligencia artificial (IA) y su aplicación específica en el campo de la robótica están presentes en este proyecto, que se centra en la creación de BuddyBot, un robot interactivo con capacidades de conducción autónoma y control por voz.

7.1. Ámbito de la inteligencia artificial

En el ámbito de la inteligencia artificial (IA), la adopción de algoritmos de aprendizaje por refuerzo ha demostrado ser crucial para la conducción autónoma. Proyectos como SoLo 12, desarrollado por PAL Robotics (2020), han consolidado la eficacia del enfoque Proximal Policy Optimization (PPO)

en la toma de decisiones autónomas. La implementación exitosa de PPO en este proyecto proporciona una referencia valiosa para la aplicación similar en el diseño de BuddyBot, respaldando así la elección de este algoritmo específico.

En paralelo, la aplicación de modelos de redes neuronales convolucionales (CNN) se fundamenta en investigaciones como el proyecto NVIDIA's End-to-End Learning for Self-Driving Cars (2016). Este proyecto ha demostrado la capacidad excepcional de las CNN para interpretar datos visuales en tiempo real, siendo esenciales para la navegación autónoma. La integración de técnicas similares en BuddyBot se basará en los principios establecidos por este proyecto de referencia, garantizando una sólida comprensión visual necesaria para la autonomía del robot.

En este contexto, se incorporarán técnicas avanzadas de conducción autónoma en BuddyBot. La aplicación de estas técnicas se apoyará en investigaciones previas como el proyecto Waymo (2016) desarrollado por Alphabet Inc., que ha demostrado la viabilidad y eficacia de los sistemas autónomos en entornos similares, consolidando así su inclusión en el repertorio técnico de BuddyBot.

7.2 Ámbito al que se aplicó la inteligencia artificial

En el ámbito de aplicación de la inteligencia artificial (IA), proyectos emblemáticos han dejado una marca significativa en la conducción autónoma y la interacción robot-humano en entornos domésticos. Ejemplos como el Stanford University's Personalized Assistant Robot (SPAR) (2016) han explorado la adaptabilidad y el aprendizaje continuo necesarios para la interacción efectiva en entornos dinámicos del hogar, brindando valiosas lecciones para la implementación de BuddyBot. Además, la investigación en robótica colaborativa, como el European Robotics Challenge (EuRoC), destaca la importancia de la seguridad y la interacción intuitiva, aspectos cruciales para la implementación de BuddyBot como un compañero de entretenimiento y seguro en el hogar.

8. Metodología

8.1. Sustento metodológico

El *Método Eisenhower* es una herramienta de gestión del tiempo que te ayuda a priorizar tareas clasificándolas según su urgencia e importancia.



Pasos:

- Anota todas tus tareas.
- Clasifícalas en cuatro cuadrantes:
 - ✓ Cuadrante 1: Tareas urgentes e importantes (Hazlas de inmediato).
 - ✓ Cuadrante 2: Tareas importantes, pero no urgentes (Prográmalas).
 - ✓ Cuadrante 3: Tareas urgentes, pero no importantes (Delega o pospóntelas).
 - ✓ Cuadrante 4: Tareas ni urgentes ni importantes (Elimínalas o pospóntelas indefinidamente).

Beneficios:

- Mejora la organización y la gestión del tiempo.
- Aumenta la productividad.
- Promueve la toma de decisiones efectiva.
- Reduce el estrés y la procrastinación.

8.2. Técnicas de recolección de datos

La obtención de datos para los modelos se realizó en dos fases:

➤ Fase 1: Recopilación de imágenes:

- Se grabaron videos desde la perspectiva inferior del robot, simulando su visión del entorno.
- Se extrajeron imágenes de los videos cada segundo utilizando un script.
- Se revisaron las imágenes para garantizar su calidad y eliminar las no útiles.
- Las imágenes de calidad se cargaron en Roboflow para su segmentación.

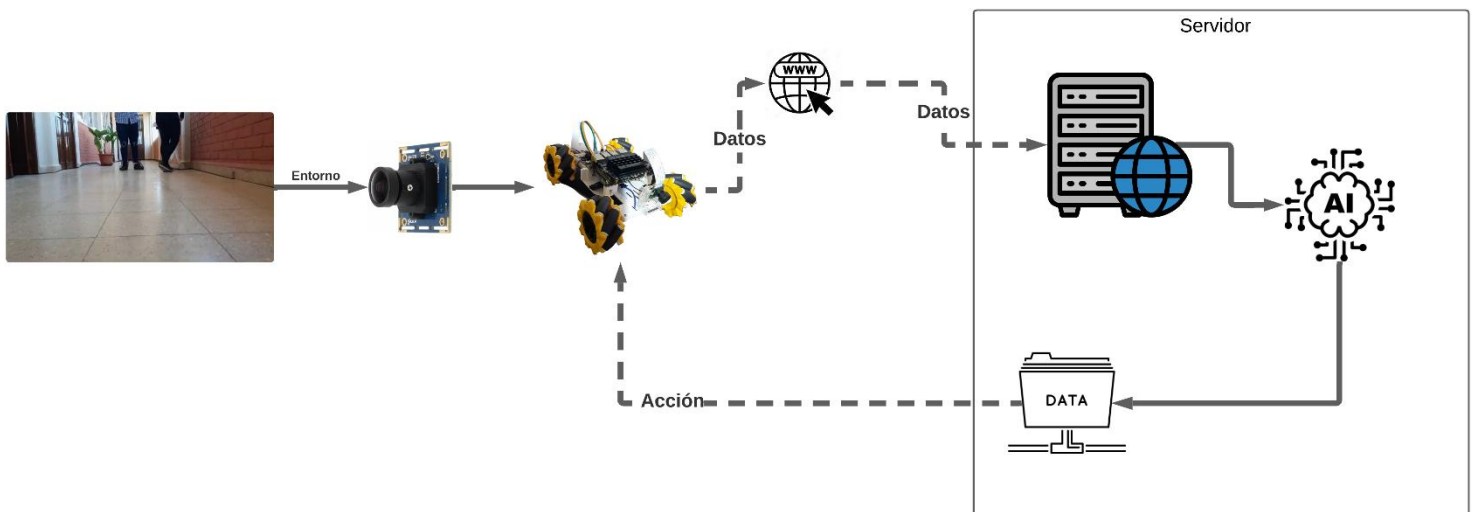
➤ Fase 2: Recopilación de datos para PPO:

- Los datos para el modelo PPO se generaron durante la ejecución del modelo.
- El modelo recibió como entrada la imagen capturada en tiempo real durante el entrenamiento.

Esta estrategia de recolección de datos permitió obtener un conjunto de datos diverso y de alta calidad para entrenar tanto el modelo UNet como el modelo PPO, lo que contribuyó al éxito general del proyecto.

8.3. Materiales & herramientas

8.3.1. Arquitectura del proyecto desarrollado.



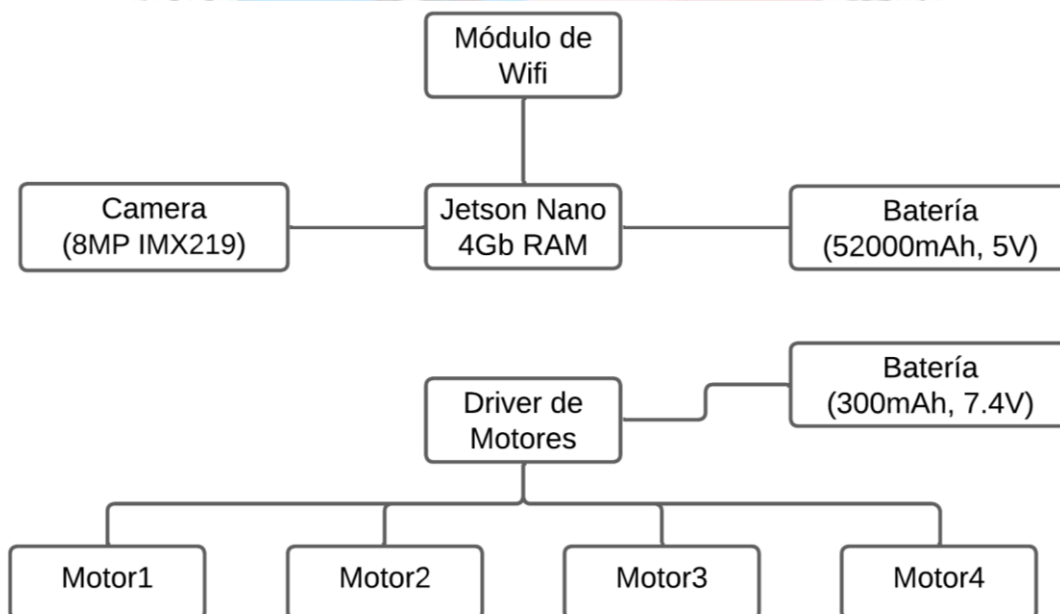
En la arquitectura del proyecto, se destacan los siguientes componentes:

- **Cámara:** Este componente proporciona la captura del entorno circundante, esencial para el suministro de datos a los modelos de inteligencia artificial. Sirve como entrada visual para la toma de decisiones del sistema.

- **Plataforma de Computación:** La Jetson Nano de NVIDIA con 4GB de RAM sirve como el núcleo computacional del sistema. Equipada con el sistema operativo Jetpack 4.6, esta placa principal alberga todos los modelos de IA implementados. Funciona como el cerebro del sistema, procesando datos y ejecutando algoritmos de forma eficiente.
- **Robot de Ejecución:** Este componente representa la parte física del sistema, responsable de llevar a cabo las acciones determinadas por los modelos de IA. Utiliza la información proporcionada por la cámara para realizar las acciones planificadas y, posteriormente, actualiza el estado del robot, cerrando así el ciclo de retroalimentación.

Esta arquitectura integra de manera efectiva la captura de datos del entorno, el procesamiento computacional y la ejecución de acciones físicas, permitiendo un funcionamiento coordinado y autónomo del sistema.

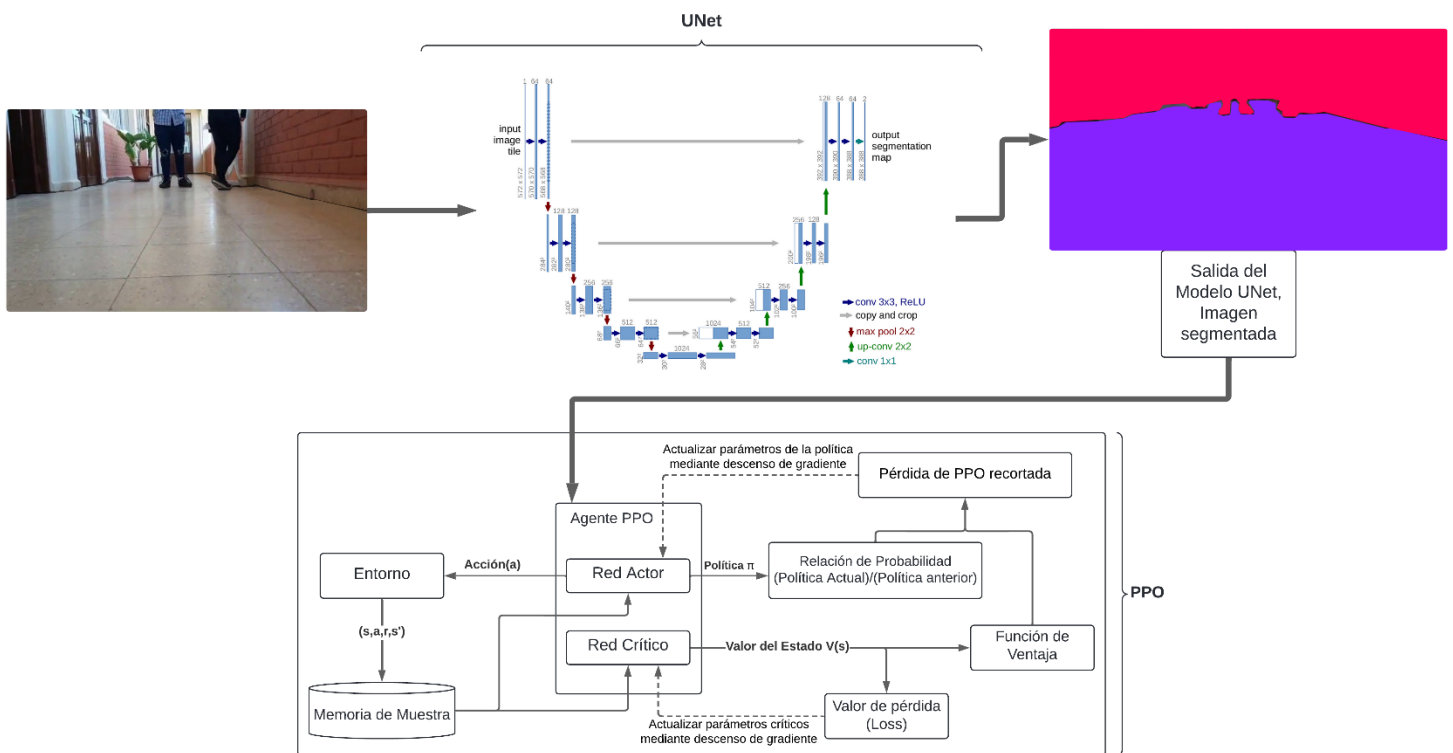
8.3.2. Esquema y descripción de componentes de hardware.



- **Placa madre:** La Jetson Nano de NVIDIA con 4GB de RAM, diseñada específicamente para ejecutar modelos de inteligencia artificial, proporciona un sólido rendimiento para el procesamiento de datos.
- **Cámara Ojo de pez (8MP IMX219):** Equipada con el potente sensor IMX219, esta cámara está especialmente diseñada para la Jetson Nano, ofreciendo un amplio ángulo de visión de 220 grados. Su versatilidad la hace ideal para una variedad de proyectos de visión artificial, vigilancia y IoT.
- **Módulo de Wifi (Intel 8265NGW M.2 Wifi-Bluetooth 5.0 compatible con Jetson):** Este módulo permite la conexión a internet y el acceso remoto al robot, facilitando la comunicación y el control desde cualquier ubicación.

- **Driver de motor:** El DC Motor + Stepper FeatherWing proporciona la capacidad de controlar los motores de manera eficiente, permitiendo el movimiento preciso del robot en diversas situaciones.
- **Batería:** La batería nano-tech de 300mAh y 7.4V proporciona la energía necesaria para alimentar el sistema, garantizando una autonomía adecuada durante las operaciones del robot.
- **Motores DC:** Los Micro Motores Pololu 10:1 HP 6V con eje extendido ofrecen potencia y precisión en el movimiento del robot, asegurando un desplazamiento suave y controlado en diferentes superficies y condiciones.

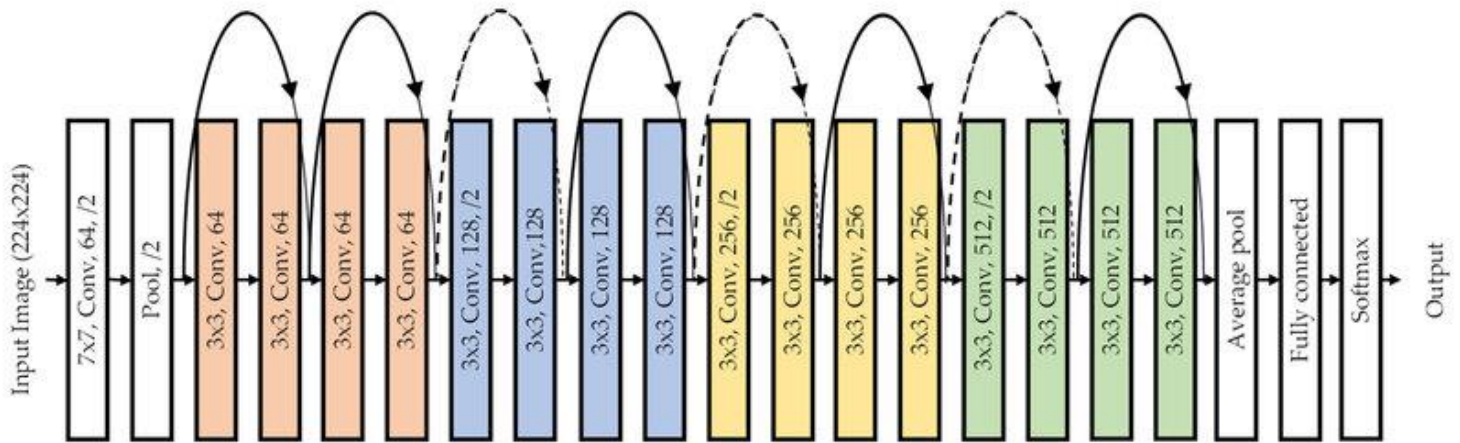
8.3.3. Esquema y descripción de modelos o componentes inteligentes



- **UNet:**
 - Este modelo recibe imágenes capturadas por la cámara con una resolución de 480x864. Al pasar por el modelo UNet, la salida es una imagen segmentada con la misma resolución.
 - Su función principal es identificar y segmentar objetos en el entorno del robot autónomo, proporcionando información crucial para la toma de decisiones.
- **Modelo PPO:**
 - Este modelo recibe la salida del modelo UNet, que previamente ha pasado por una capa de MaxPooling para reducir la imagen a una resolución de 60x108, conservando sus características más importantes.

- La máscara resultante se utiliza como entrada para el modelo PPO, que genera la acción que el agente tomará en función de la información del entorno segmentado.

➤ **Red Neuronal Actor:**



- Esta red neuronal es una ResNet18 a la que se le aplicará transfer learning para mejorar la toma de decisiones del modelo.
- Su objetivo es aprender y representar las políticas de acción del agente, permitiendo tomar decisiones óptimas en función de las entradas del entorno.

➤ **Red Neuronal Critic:**

- Esta red neuronal es una CNN (Convolutional Neural Network) que califica la calidad de las acciones tomadas por el agente.
- Su salida es un valor que representa la evaluación de la acción realizada, lo que permite retroalimentar el proceso de aprendizaje y mejorar la efectividad del sistema en la conducción autónoma.

8.3.4. Valores de parámetros e hiperparámetros aplicados

Parámetros:

UNet	
layers	18
parameters	11173962
PPO (Proximal Policy Optimization)	
layers	22
parameters	13284352

Hiperparámetros:

<i>UNet</i>	
epochs	30
Batch_size	6
Image_size	(480, 864)
device	“cuda”
workers	24
pretrained	True
optimizer	Adam
Learning_rate	0.0003
<i>PPO (Proximal Policy Optimization)</i>	
Learning_rate	0.0003
learn_iters	4
policy_clip	0.2
optimizer	Adam()
Image_size	(60, 108)
n_games	30
best_score	-infinity
Minibatch size	4

8.3.5. Especificaciones técnicas.

El modelo se entrenó utilizando una GPU RTX3060 con 12GB de RAM durante un período de 9 horas para el modelo UNet destinado a la segmentación. En cuanto al modelo PPO, debido a limitaciones de tiempo para pruebas en un simulador, se optó por realizar el entrenamiento directamente en el robot físico. A pesar de la complejidad adicional, se logró entrenar el modelo en el entorno físico. Además, implementamos un servidor en el que se alojan todos los modelos, permitiendo que BuddyBot consuma sus servicios desde el servidor. Esta decisión se tomó debido a que los modelos son demasiado grandes para ser soportados por la Jetson Nano.

8.3.6. Lenguajes de programación, frameworks, entre otros.

Para el desarrollo del proyecto, se emplearon diversas herramientas y tecnologías. Se utilizó el lenguaje de programación Python como base, junto con los frameworks PyTorch y OpenCV (cv2) para la implementación y entrenamiento de los modelos de inteligencia artificial. Además, se emplearon librerías adicionales como:

- Pytorch (2.3.0)
 - OpenCV(4.6.0)
 - Adafruit_MotorHAT (1.4.0)
 - Traitlets (4.3.3)
 - Requests (2.32.2)
- Numpy (1.24.3)
 - Matplotlib (3.8.4)
 - Pillow (9.3.0)
 - Torchvision (0.18.0)
 - Pycocotools (2.0.6)
- Tqdm (4.66.4)
 - Pandas (2.2.1)
 - Flask (3.0.3)
 - Tkinter (8.6.14)

8.4. Plan de trabajo



9. Resultados

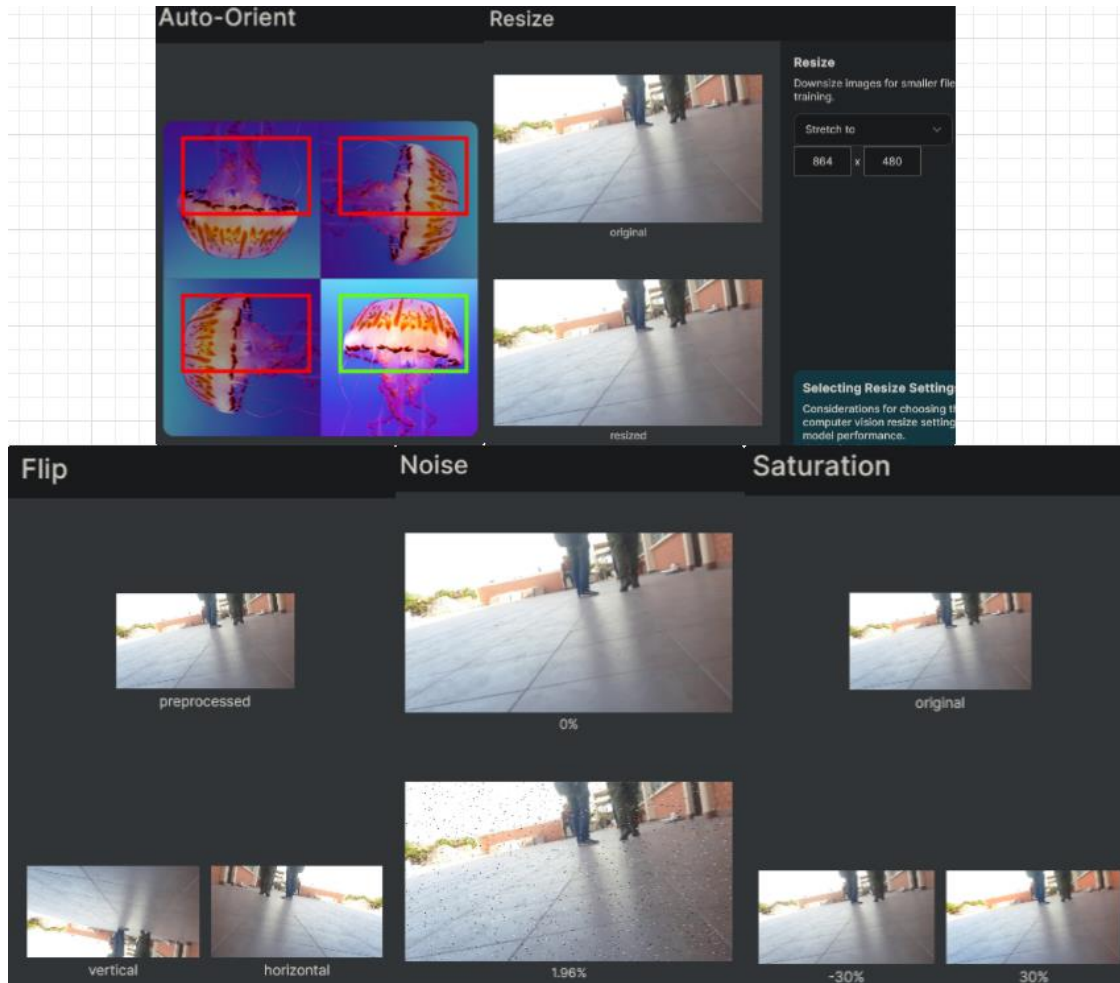
9.1. Dataset

9.1.1. Descripción y preprocesamiento realizado UNet:

Se realizó un exhaustivo preprocesamiento de los videos grabados, extrayendo imágenes por segundo mediante un script en Python. Se llevó a cabo una revisión minuciosa para descartar imágenes borrosas. Además, se redujo la resolución de las imágenes de 4K a 480x864 para mejorar el rendimiento del modelo, asegurando la calidad y eficiencia de los datos de entrada.

Preprocesamiento	Orientación automática: aplicada Cambiar tamaño: estirar a 480x864
------------------	---

Data Augmentations	Data Augmentation por Imagen: 3 Voltear: Horizontal, Vertical Saturación: Entre -30% y +30% Ruido: Hasta el 1,96% de los píxeles
--------------------	---



PPO (Proximal Policy Optimization):

Durante el proceso de entrenamiento del modelo PPO, los datos se generan dinámicamente en el momento. Para optimizar la eficiencia del modelo, se realizó un preprocesamiento específico en los datos de entrada. Inicialmente, el modelo UNet produce una máscara con una resolución de 480x864, que resulta ser demasiado grande para el modelo PPO. Para solucionar esto, se aplicó una técnica de MaxPooling a la máscara, lo que permitió extraer las características más relevantes de la imagen y reducirla a una resolución de 60x108. Esta versión reducida y simplificada de la máscara se utilizó como entrada para el modelo PPO, optimizando así el proceso de entrenamiento y mejorando su rendimiento.

9.1.2. Conjunto de entrenamiento, evaluación y validación

UNet:

El conjunto de datos se dividió en dos conjuntos distintos para entrenamiento y prueba. El conjunto de entrenamiento representa el 89% del total de imágenes, con un total de 5292 imágenes. Para la prueba del modelo, se destinaron un 11% de imágenes a cada conjunto, lo que equivale a 605 imágenes para el conjunto de pruebas. En total, se cuenta con 5897 imágenes disponibles para el entrenamiento y prueba del modelo.

PPO (Proximal Policy Optimization):

Dado que el modelo PPO es de aprendizaje por refuerzo, solo utiliza un conjunto de entrenamiento. En este caso, las salidas del modelo UNet, es decir, las máscaras resultantes de la segmentación, sirven como los datos de entrada. Estas máscaras representan el entorno y son la única información necesaria para que el modelo PPO aprenda a tomar decisiones de acción óptimas. Por lo tanto, el entrenamiento del modelo PPO se basa exclusivamente en este conjunto de datos de entrenamiento.

9.1.3. Técnicas, criterio y/o métodos aplicados para la conformación de los conjuntos de datos de entrenamiento, prueba y validación.

- **División Aleatoria:** Dividir aleatoriamente el conjunto de datos en porcentajes predefinidos para entrenamiento y prueba. Esto garantiza una distribución equitativa y representativa de los datos en cada conjunto.
- **Estratificación:** Asegurar que cada conjunto de datos tenga una distribución similar de clases o características importantes para evitar el sesgo. Esto es especialmente útil en conjuntos de datos desequilibrados.

Equilibrio de clases

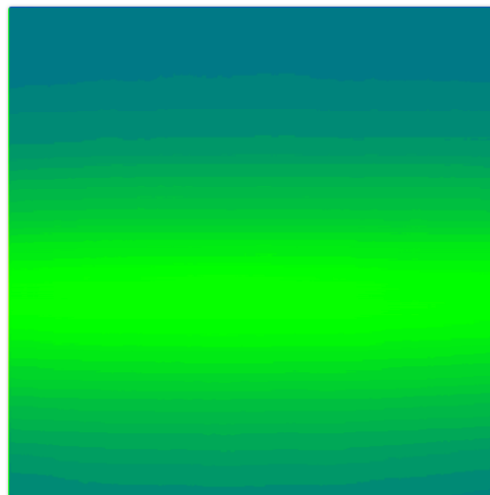
un obstaculo
camino

2,140
1,976



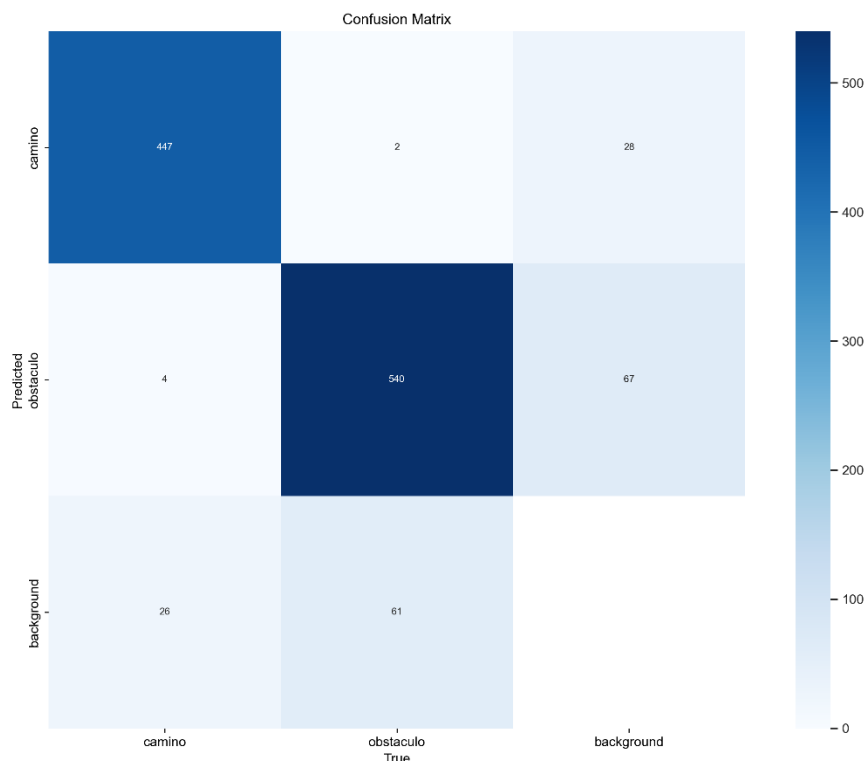
- **Mapa de calor de anotación:**

Mapa de calor de anotación

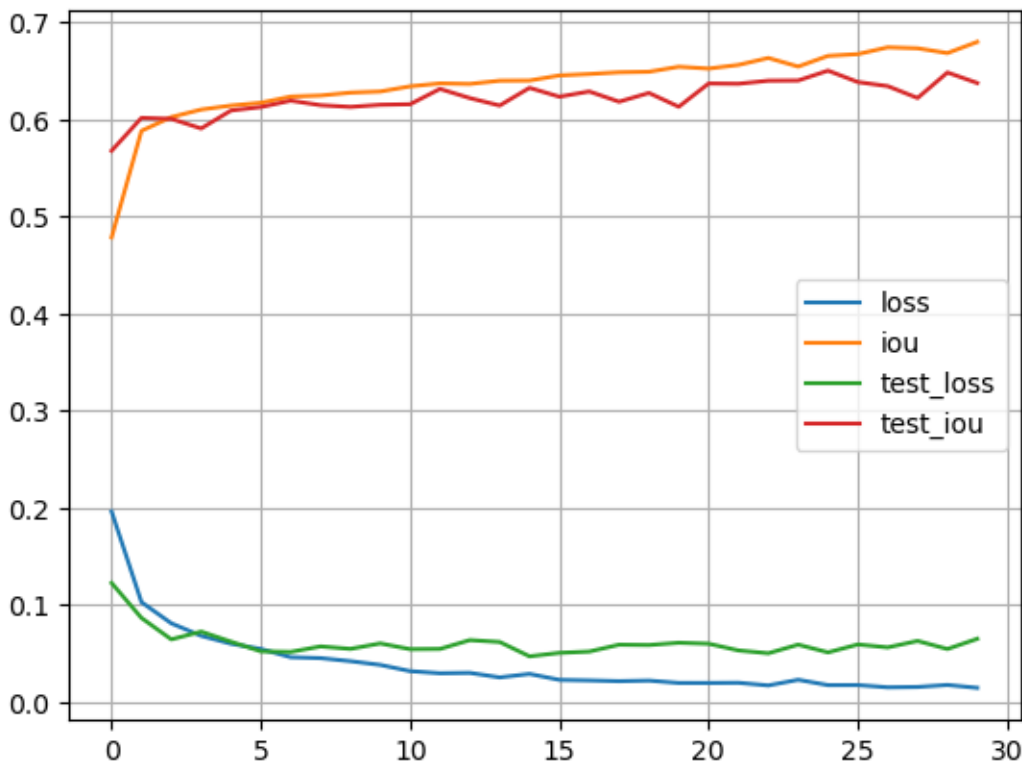


9.2. Resultados de entrenamiento y prueba (Métricas de rendimiento)

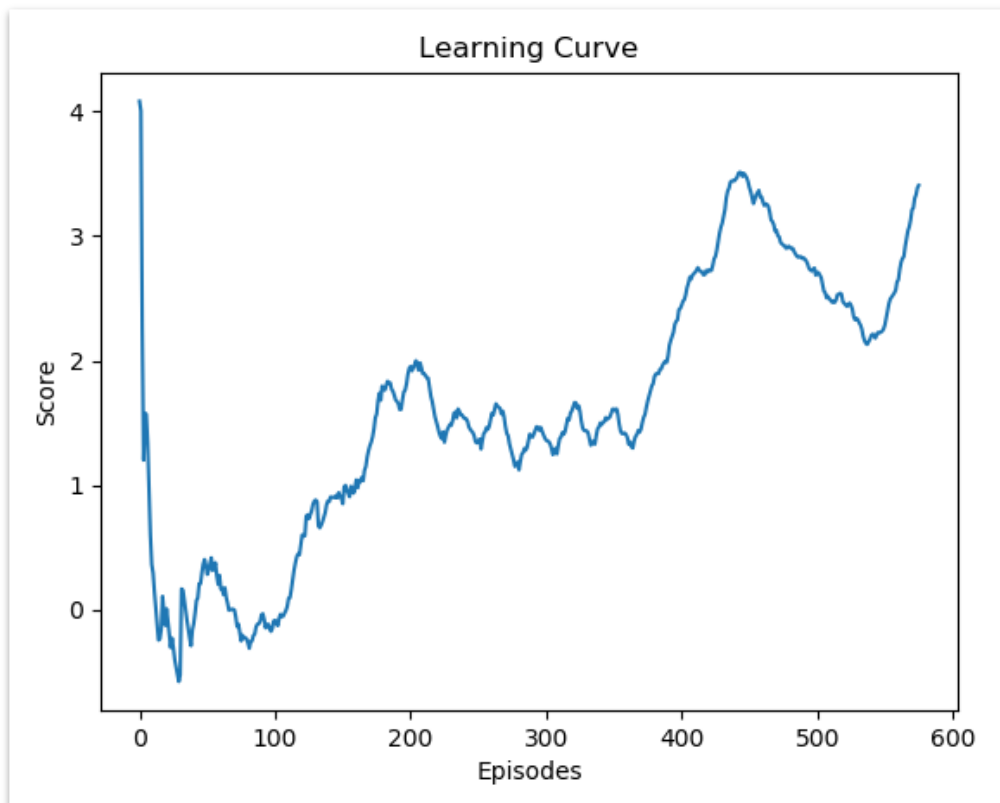
9.2.1. Matriz de confusión



9.2.2. Perdidas y Presición (UNet)

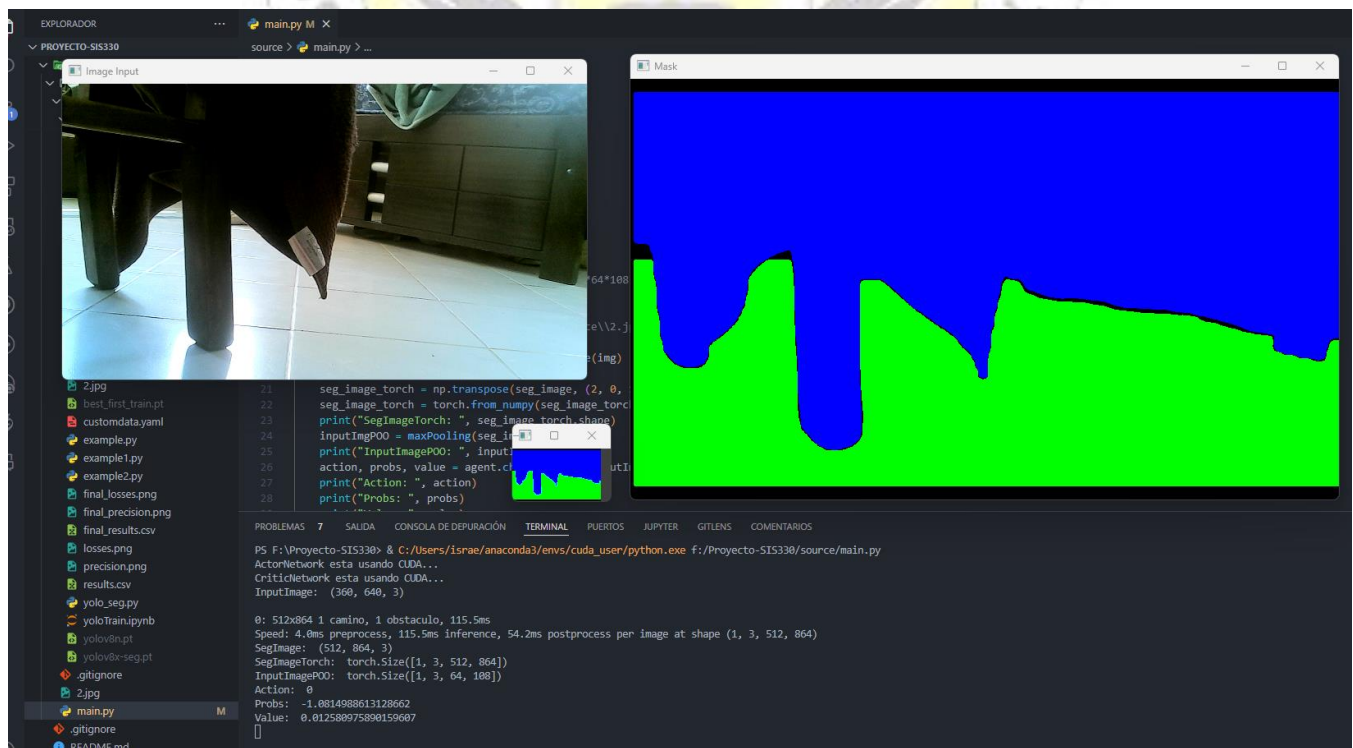


9.2.3. Curva de aprendizaje (PPO)



9.3. Resultados de aplicación

Resultados de los prototipos de los modelos implementados:



El cuadro de entrada de imagen, denominado "Image Input", representa la imagen adquirida por la cámara. Esta imagen se introduce en el modelo de segmentación UNet. Posteriormente, la máscara resultante se somete a un proceso de reducción de tamaño mediante un modelo de MaxPooling, con el objetivo de transformarla de una resolución de 480x864 a 60x108 píxeles. Esta versión reducida de la máscara, conocida como "minimáscara", es la entrada para el modelo de aprendizaje por refuerzo. El modelo de aprendizaje por refuerzo genera una acción, así como la probabilidad logarítmica asociada y el valor de dicha acción. Estos resultados son cruciales para el proceso de entrenamiento del modelo.

10. Conclusiones

El proyecto de conducción autónoma de BuddyBot ha alcanzado importantes logros, contribuyendo al avance en el campo de la robótica y la inteligencia artificial. A través de la investigación y la implementación de modelos avanzados de aprendizaje profundo, como Proximal Policy Optimization (PPO) y Convolutional Neural Networks (CNN), se logró mejorar significativamente la capacidad de BuddyBot para la navegación autónoma.

La combinación de algoritmos de aprendizaje por refuerzo para la detección de obstáculos y la navegación autónoma, junto con las funciones de reconocimiento de voz para una interacción personalizada, ha resultado en el desarrollo exitoso de un prototipo funcional de BuddyBot. Este prototipo representa un compañero robot interactivo y versátil, capaz de adaptarse al entorno y satisfacer las necesidades del usuario de manera efectiva.

Las pruebas exhaustivas realizadas para validar la efectividad de los modelos de inteligencia artificial en BuddyBot han demostrado su rendimiento óptimo en conducción autónoma. Sin embargo, se identificaron áreas de mejora durante las pruebas, lo que sugiere la necesidad de ajustes continuos y refinamientos en los modelos para garantizar su óptimo funcionamiento en una variedad de escenarios y condiciones del mundo real.

11. Recomendaciones

- **Continuar la Optimización de los Modelos:** Se recomienda continuar refinando y optimizando los modelos de inteligencia artificial implementados en BuddyBot, como el Proximal Policy Optimization (PPO) y las Convolutional Neural Networks (CNN), para mejorar aún más su rendimiento y robustez en una variedad de situaciones y entornos.
- **Explorar la Integración de Sensores Adicionales:** Considerar la integración de sensores adicionales, como lidar o ultrasonidos, para mejorar la percepción del entorno de BuddyBot y aumentar su capacidad de detección y navegación autónoma en condiciones adversas o poco comunes.
- **Realizar Pruebas en Diversos Escenarios:** Es importante llevar a cabo pruebas exhaustivas en una variedad de escenarios del mundo real para evaluar la capacidad de BuddyBot para adaptarse a diferentes condiciones de iluminación, superficies de carreteras y situaciones de tráfico.
- **Implementar Funcionalidades de Seguridad Avanzada:** Incorporar características de seguridad avanzada, como sistemas de frenado de emergencia o detección de colisiones, para garantizar la seguridad tanto de BuddyBot como de los peatones y otros vehículos en la carretera.

- **Desarrollar Capacidades de Aprendizaje Continuo:** Explorar la implementación de capacidades de aprendizaje continuo en BuddyBot para permitir que el robot mejore su desempeño y se adapte a nuevas situaciones y desafíos a lo largo del tiempo.
- **Fomentar la Interacción Usuario-Robot:** Continuar investigando y desarrollando funciones de reconocimiento de voz y otras formas de interacción humano-robot para mejorar la experiencia del usuario con BuddyBot y promover una comunicación más natural y fluida.
- **Mantener la Actualización y Mantenimiento:** Establecer un plan de mantenimiento regular para garantizar que el hardware y el software de BuddyBot se mantengan actualizados y funcionen de manera óptima a lo largo del tiempo. Esto incluye la corrección de errores, la aplicación de parches de seguridad y la incorporación de nuevas funcionalidades según sea necesario.

12. Referencias bibliográficas

- [1] Zárata Gómez, A. A., Fajardo Correa, J. J. y Guerrero Rueda, Y. A. (2020). Influencia de la tecnología en la interacción social adolescente. Documentos de Trabajo Areandina (1). Fundación Universitaria del Área
- [2] Delgado Villalobos, Sophia & Qiu, Cheng & Cordero, Leticia & Sibaja, Mauricio. (2022). Impacto del sedentarismo en la salud mental.. Revista Ciencia y Salud Integrando Conocimientos. 6. 10.34192/cienciaysalud.v6i1.404.
- [3] "El impacto de las redes sociales en la salud mental de los jóvenes" (2021) Disponible en: <https://www.periodicolaverdad.com/el-impacto-de-las-redes-sociales-en-la-salud-mental-de-los-jovenes-el-caso-de-la-demanda-contra-meta/>
- [4] "La salud mental en la era digital" (2022) Disponible en: <https://www.mhe-sme.org/wp-content/uploads/2023/04/LASALU2.pdf>
- [5] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (n.d.). Proximal Policy Optimization Algorithms. OpenAI. [Correo electrónico: joschu, filip, prafulla, alec, oleg]@openai.com.
- [6] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., & Zhokhov, P. (2017). OpenAI Baselines. GitHub repository. Recuperado de: <https://github.com/openai/baselines>
- [7] Moralejo Piñas, J. (2019). Deep Reinforcement Learning aplicado a la Conducción Autónoma. Trabajo Fin de Grado, Grado en Ingeniería Informática, Leganés.
- [8] Salvador Ramos, B. (2020). Aprendizaje automático para conducción autónoma. Tesis de Máster en Ingeniería Electrónica, Universidad de Zaragoza.
- [9] Reis, D., Kupec, J., Hong, J., Daoudi, A. (s.f.). Real-Time Flying Object Detection with YOLOv8. Georgia Institute of Technology. (2023)
- [10] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany. Retrieved from <http://lmb.informatik.uni-freiburg.de/>