

# Conducción autónoma sobre plataforma real y simulada con seguimiento de carril e identificación de señales de tráfico y peatones mediante redes neuronales

Álvaro Mariscal Ávila

[a.mariscal.2018@alumnos.urjc.es](mailto:a.mariscal.2018@alumnos.urjc.es)



Trabajo fin de grado

5 de julio de 2022

1. Nombre
2. Título: Conducción autónoma sobre plataforma real y simulada con seguimiento de carril e identificación de señales de tráfico y peatones mediante redes neuronales.
3. El trabajo se enmarca en la robótica y la visión artificial.



(CC) Julio Vega

*Este trabajo se entrega bajo licencia CC BY-NC-SA.  
Usted es libre de (a) compartir: copiar y redistribuir el material en  
cualquier medio o formato; y (b) adaptar: remezclar, transformar  
y crear a partir del material. El licenciador no puede revocar estas  
libertades mientras cumpla con los términos de la licencia.*

1. La presentación esta dividida en cinco partes.

1 Introducción

2 Objetivos

3 Plataforma de desarrollo

4 Sistema de conducción autónoma

5 Conclusiones

1. Para empezar tenemos la introducción para hablar acerca del contexto.

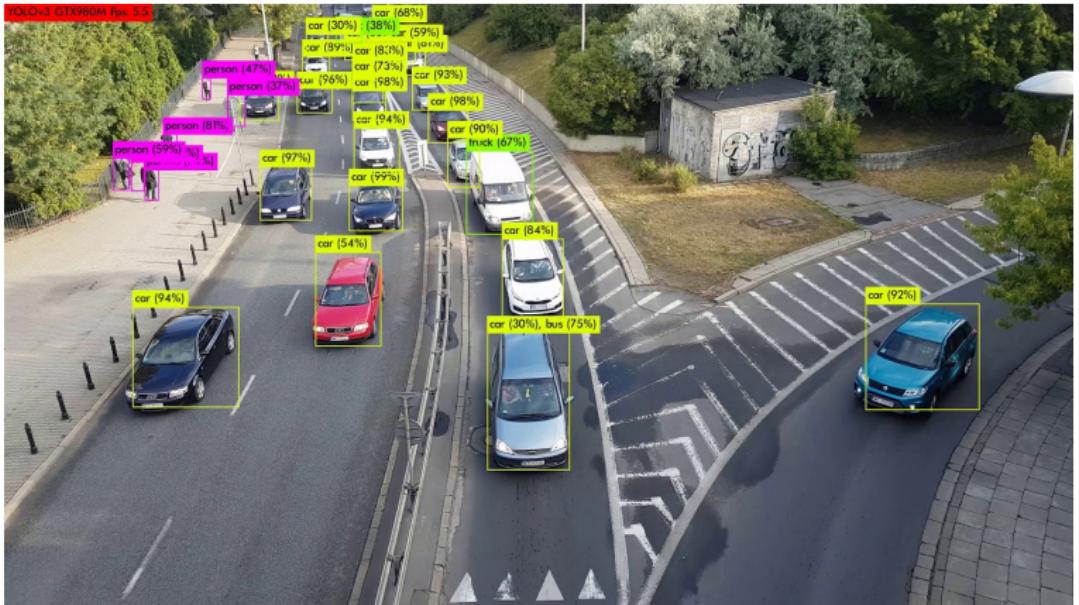
# *Introducción*

# Inteligencia artificial

- Entender y emular el comportamiento humano.
- Dotar a sistemas de cierta **inteligencia** y de la capacidad de **aprender**.
- Visión artificial, aprendizaje automático o aprendizaje profundo.

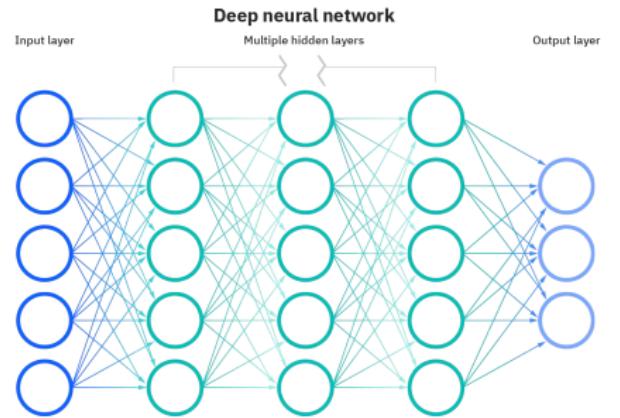
1. Inteligencia artificial: es la disciplina .....
2. Dotar a sistemas entre los que se encuentran los robots ....
3. Multitud de ramas que parten de la IA .....

# Visión artificial



1. Visión artificial utilizando sensores como las cámaras permite obtener una información muy completa del entorno.
2. Tiene multitud de aplicaciones entre las que se encuentra la detección de objetos.
3. Para ello es necesario técnicas como Deep Learning o aprendizaje profundo.

# Deep Learning



1. Deep Learning se basa en el uso de redes neuronales artificiales.
2. Estas están inspiradas en el cerebro humano.
3. En el caso de la abstracción al mundo de la computación, una red está formada por una capa de entrada, una de salida y varias internas u ocultas.
4. Estas redes se entrenan previamente mediante lo conocido como datasets. Que son grandes volúmenes de datos, ya sean imágenes, videos, sonidos etc.
5. Neocognitrón. Además, decir que no existe límite claro para definir número de capas que debe tener una red neuronal para considerarla Deep Learning.

- Neocognitrón (1979): Red neuronal con 5 o 6 capas para reconocer caracteres japoneses.

# Vehículos autónomos



1. En cuanto al contexto específico en el que se enmarca el trabajo, encontramos los vehículos autónomos que se permiten circular sin conductor.
2. Si bien es cierto que aún no vemos este tipo de vehículos circulando por las ciudades, mucha de la tecnología necesaria ya está disponible en los vehículos actuales
3. Existen diferentes niveles de autonomía. En el caso de la fotografía es un camión autónomo de la marca Volvo con nivel de autonomía 5, es decir, automatización completa en cualquier situación. Actualmente se están desarrollando pruebas en el puerto de Gotenburgoooooo
4. Y además, decir que la posibilidad de automatizar transportes de mercancías, sería un gran avance por las grandes distancias que recorren, incluyendo en algunos casos transporte internacional durante grandes intervalos de tiempo

# Autonomous Mobile Robots, AMRs



- Sucesores de los AGVs.

1. Son aquellos capaces de navegar por entornos dinámicos, conviviendo con humanos y sabiendo sobreponerse a situaciones para las que no habían sido programados explícitamente.
2. Son los sucesores de los AGVs. Estos requieren una cierta infraestructura dependiendo del tipo de guiado, ya sea filo-guiados, a través de pintura o a través de cualquier otra técnica que haga que ese vehículo solo pueda funcionar cuando se conoce la infraestructura previa que estará presente en el entorno de trabajo.
3. Kiva Systems: comprada por Amazon en 2012. Permiten automatizar sus almacenes en tareas de logística a nivel interno, maximizando la productividad y el almacenamiento, tanto en profundidad como en altura, y obviamente minimizando el coste en personal..

# *Objetivos*

## Descripción del problema

### 1. Descripción.

- Desarrollar un **coche autónomo** capaz de circular por un **circuito** en un entorno dinámico, interactuando con objetos propios de una ciudad en dos entornos distintos:
  - Entorno **simulado** con *Gazebo*.
  - Entorno **real** usando un robot con *Jetson Nano*.
- En ambos entornos se han de cumplir dos subobjetivos:
  - Seguimiento de **carril**
  - Detección de **objetos**

# Requisitos

## 1. Requisitos.

- El sistema operativo utilizado será *GNU/Linux*, concretamente la distribución *Ubuntu 18.04 LTS*.
- El entorno simulado requerirá la presencia de una tarjeta gráfica dedicada: *NVIDIA* y *CUDA*.
- El entorno real requerirá un robot con la placa de desarrollo *NVIDIA Jetson Nano*, ya que esta es una de las placas con *GPU* más económicas.
- El lenguaje de programación utilizado será *Python*.

# *Plataforma de desarrollo*

# NVIDIA Jetson Nano

## 1. NVIDIA Jetson Nano.

- Placa de desarrollo de **bajo coste** con **GPU** dedicada.
- Arquitectura **Aarch64**, soporte para **GNU/Linux** y puertos **GPIO**.



# Componentes



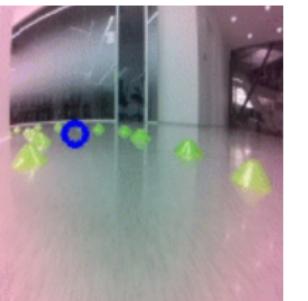
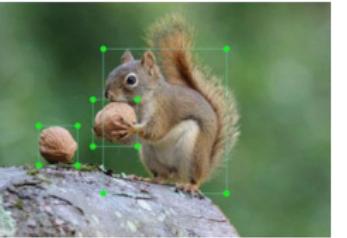
1. Componentes.
2. Motores de bajo coste sin odometría.
3. Controlador de motores. Limitaciones motores a pares.
4. Batería 10000mAh para alimentar Jetson Nano.
5. Batería 7.4V para alimentar motores.
6. Cámara USB como sensor principal.
7. Adaptador Wi-Fi para conexión mediante SSH con el robot.
8. Chasis muy usado en proyectos relacionados con Arduino.

## Software



1. Software.
2. Python.
3. FreeCAD.
4. Blender.
5. Gazebo.
6. ROS.
7. PyQt.

## Software relacionado con visión



1. Software relacionado con visión.
2. OpenCV.
3. YOLO.
4. LabelIMG.
5. Darknet.
6. JetRacer.

1. Una vez descritos los objetivos y las plataformas utilizadas veamos el desarrollo en sí del proyecto.

# *Sistema de conducción autónoma*

## Modelo de la ciudad en Gazebo

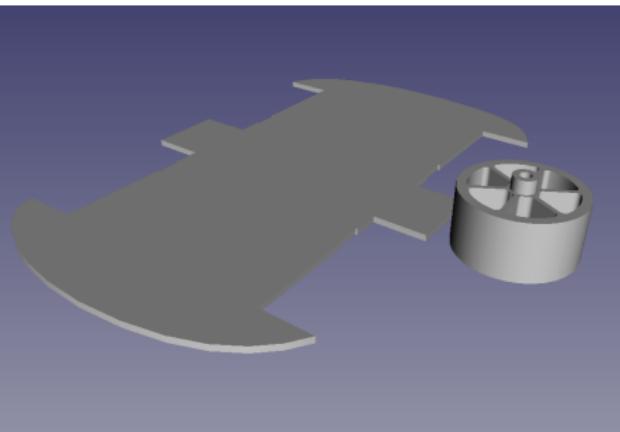
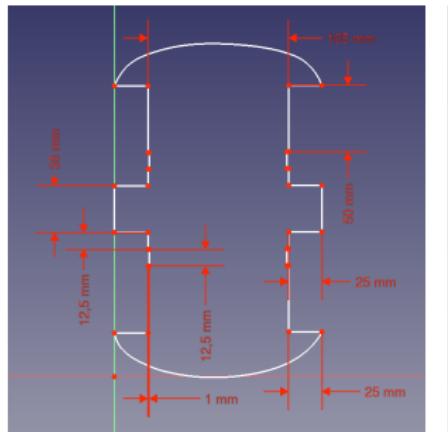
1. Modelo.



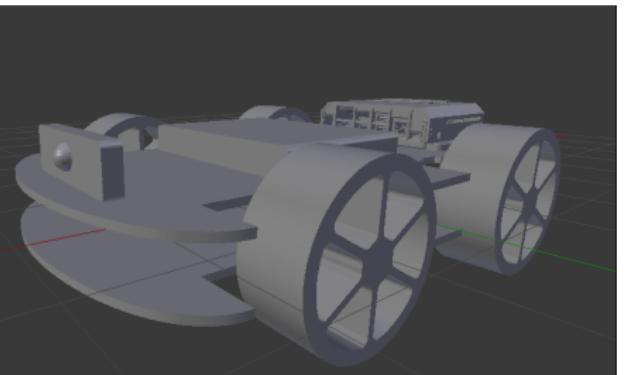
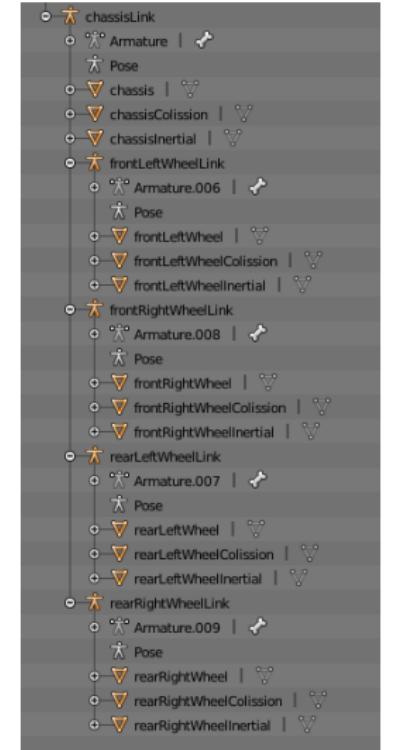
- Reproducible en el entorno real
- Semáforo y peatón dinámico

# Modelo del vehículo en *FreeCAD*

## 1. Modelo del coche autónomo.



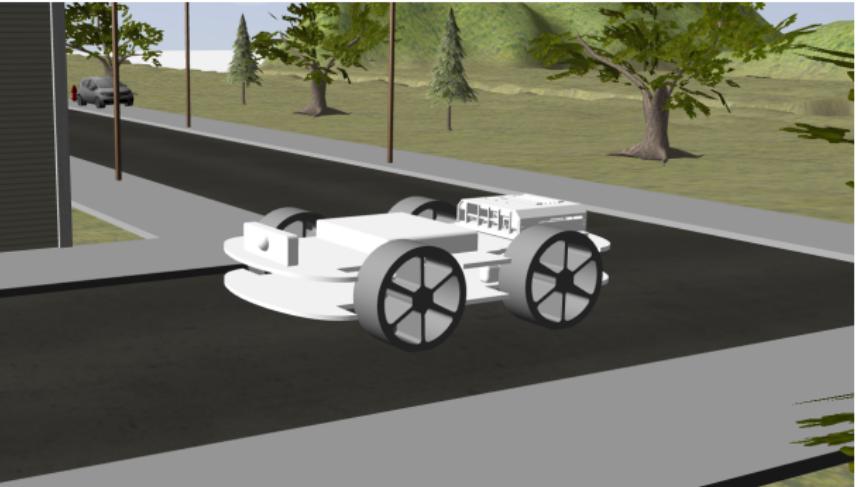
# Modelo y jerarquía del vehículo en *Blender*



## 1. Modelo y jerarquía en Blender.

## Modelo del vehículo en Gazebo

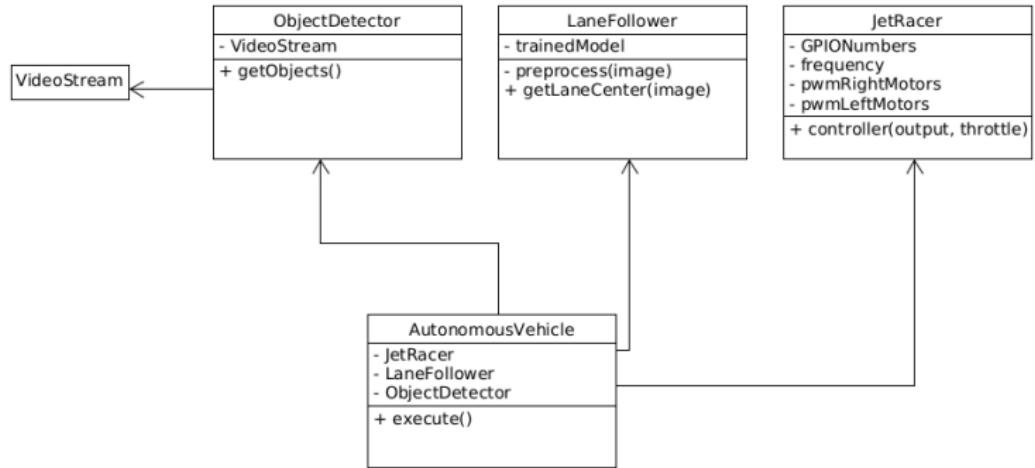
1. Modelo del vehículo en Gazebo.



- *Plugins* para movimiento y cámara conectados con *ROS*

# Diagrama de clases

## 1. Diagrama de clases.



## Seguimiento de carril

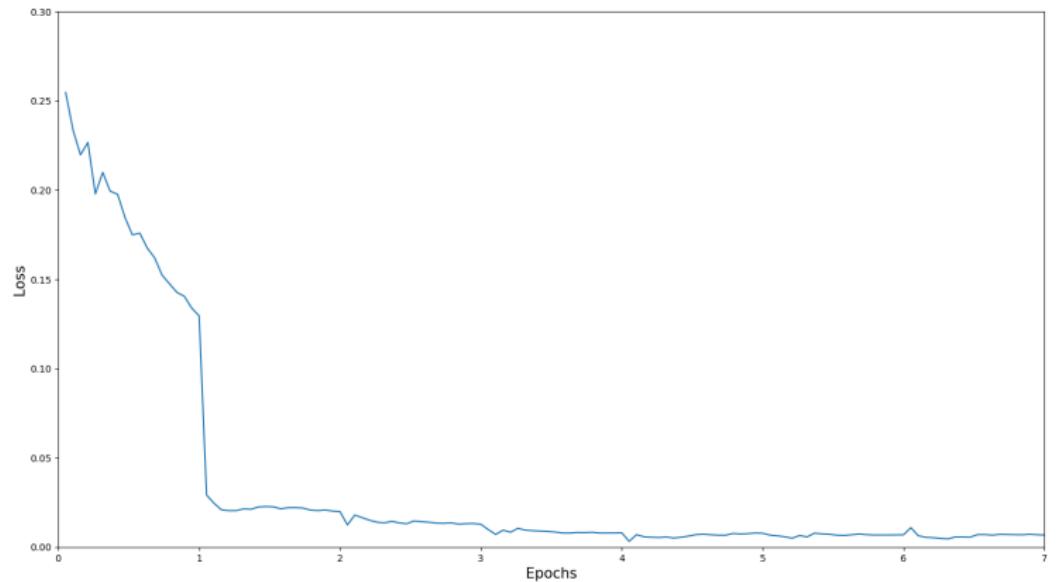
1. PyTorch.



- Red *ResNet-18* preentrenada: convolucional de 18 capas con **bloques residuales**.
- *Notebook* de *Jupyter*.
- Guardado de cada imagen: *x\_y\_identificador\_unico.jpg*.
- Dataset de 200 imágenes incluyendo situaciones **difíciles**.

## Entrenamiento red seguimiento de carril

### 1. Entrenamiento red seguimiento de carril.



- En cada *epoch* se calcula el error cuadrático medio (*loss*).
- Propagación hacia atrás de los errores desde la capa de salida hasta la primera capa.

## Salida red seguimiento de carril

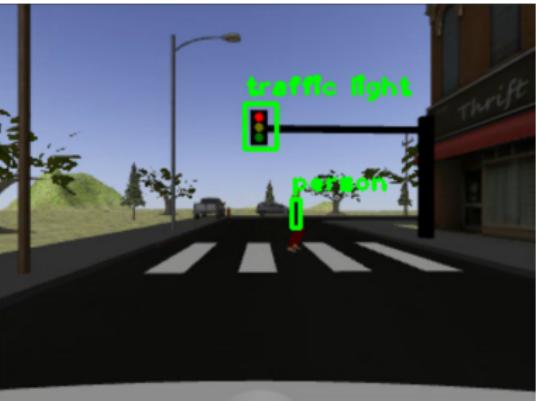
1. Entrenamiento red seguimiento de carril.



- 224 píxeles.
- Entorno experimental: autómata.

## Detección de objetos

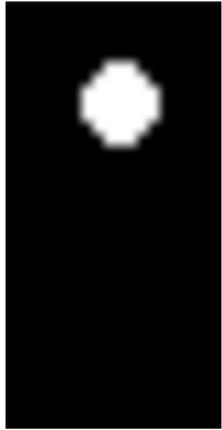
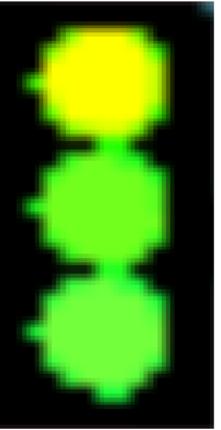
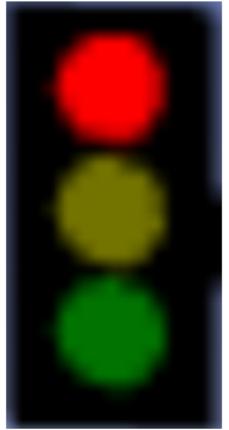
### 1. Detección de objetos.



- *YOLO V3 Tiny*.
- Objetos muy *genéricos*: alta probabilidad de detección.
- *Bounding Box*: clase y probabilidad.

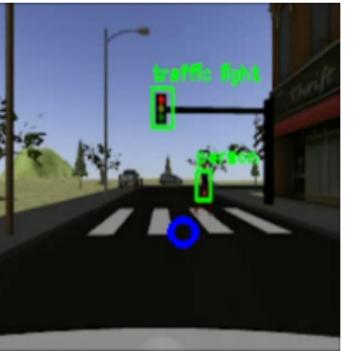
## Detección de semáforo

1. Detección de semáforo.



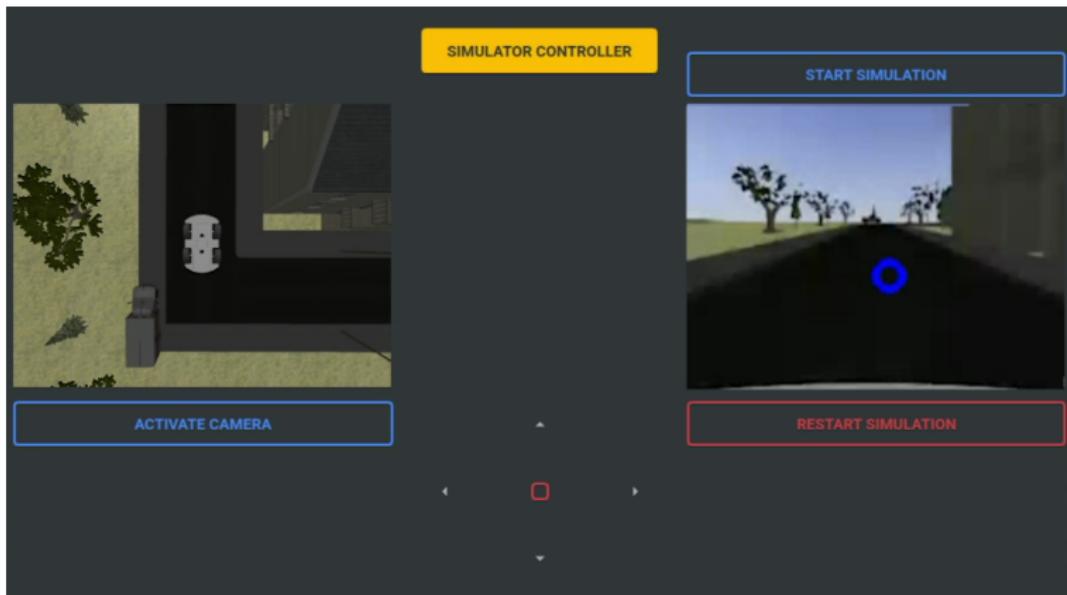
## Ejecución en el simulador

1. Ejecución en el simulador.



# Interfaz de usuario

## 1. Interfaz de usuario.



# Entorno real

## 1. Entorno real.



## Círculo inicial

### 1. Circuito inicial.



- Entrenamiento con luz *artificial*.

## Círculo con objetos

1. Circuito con objetos.



## Dataset objetos reales

### 1. Dataset objetos reales.



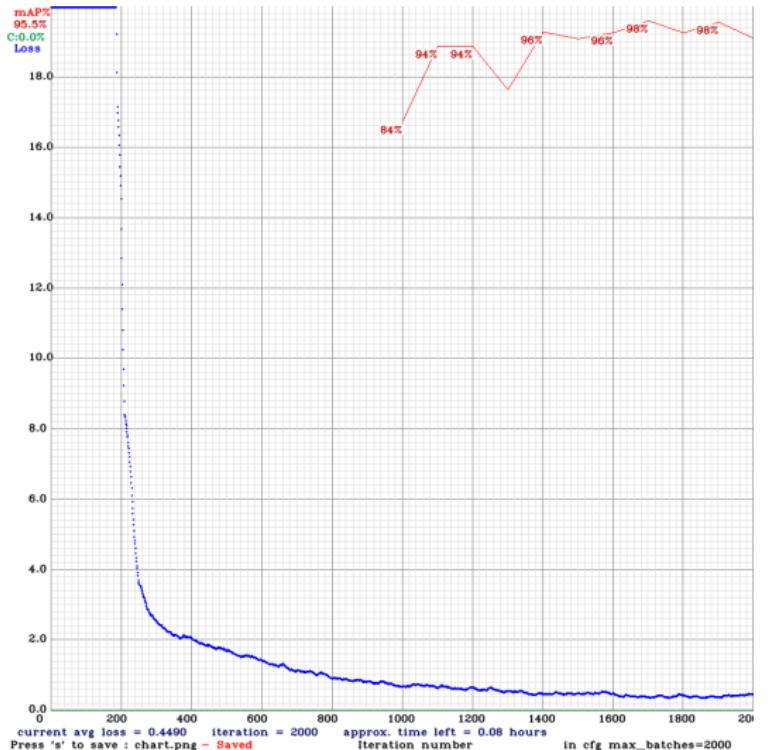
## Entrenamiento red de detección de objetos

### 1. Entrenamiento red de detección de objetos.



- Portátil con *NVIDIA MX330*  $\simeq 95^\circ C$ .
- 2000 iteraciones  $\simeq 3$  horas.
- Únicamente con *dataset* propio.

# Entrenamiento red de detección de objetos



## 1. Entrenamiento red de detección de objetos.

## Comparación probabilidad de detección

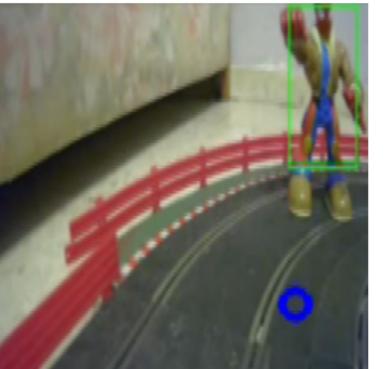
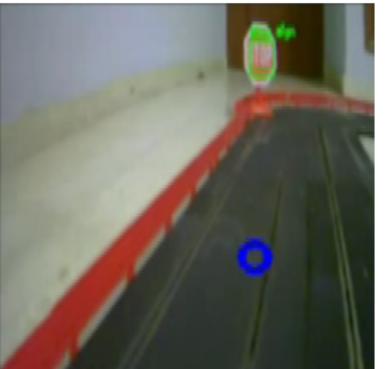
- Imagen **desconocida** para ambas redes.



1. Comparación probabilidad de detección.
2. Marca la diferencia entre detectar o no al peatón cuando el robot está en movimiento.

## Ejecución en el entorno real

1. Ejecución en el entorno real.



# *Conclusiones*

1. Para acabar esta presentación, vamos a repasar lo hecho, unas breves conclusiones y las líneas futuras.

# Conclusiones

## 1. Conclusiones.

- Para lograr los objetivos se han utilizado dos **redes neuronales**:
  - Seguimiento de carril: librería *JetRacer* que implementa una red *residual ResNet-18* combinada con un **controlador**.
  - Detección de objetos: *framework Darknet* que permite ejecutar una red *convolucional YOLO V3 Tiny* previamente entrenada mediante un **dataset** propio con los objetos reales para aumentar la fiabilidad.
- Todo ello ha sido combinado en dos paquetes *ROS*, diferenciando entorno simulado y real.
- Limitaciones: **ángulo** de la cámara y **resolución** de imagen en las redes neuronales.

# Líneas futuras



1. Líneas futuras: mayor dataset, túnes, bosques, todo entorno donde exista algo similar a un camino a seguir. También sería interesante obtener los límites en izquierda y derecha del carril y no solo el punto central.

# Conducción autónoma sobre plataforma real y simulada con seguimiento de carril e identificación de señales de tráfico y peatones mediante redes neuronales

Álvaro Mariscal Ávila

[a.mariscal.2018@alumnos.urjc.es](mailto:a.mariscal.2018@alumnos.urjc.es)



Trabajo fin de grado

5 de julio de 2022

1. Vídeo.
2. Eso sería todo. Muchas gracias por su atención.