

CONDUCCIÓN AUTÓNOMA DE UN VEHÍCULO EN SIMULADOR MEDIANTE APRENDIZAJE EXTREMO A EXTREMO BASADO EN VISIÓN



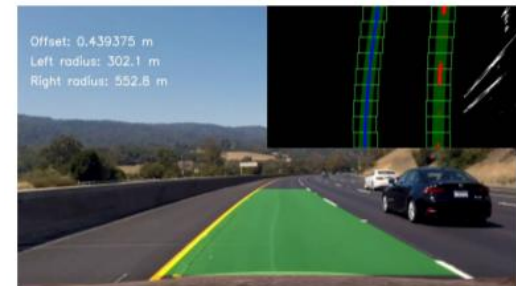
Vanessa Fernández Martínez
v.fernandezmarti@alumnos.urjc.es

25 de julio de 2019

- Introducción
- Objetivos
- Infraestructura
- Infraestructura desarrollada
- Redes de clasificación
- Redes de regresión
- Conclusiones

Conducción autónoma

- Vehículos con sensores
- Aprendizaje de normas de circulación
- Redes neuronales
- Reducción de tasa de accidentes



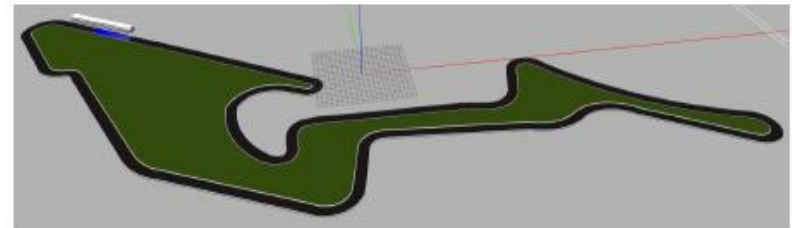
Estudio de redes neuronales para conducción autónoma en simulación

- Aplicación de control visual.
- Creación de bases de datos propias.
- Estudio de redes neuronales convolucionales de clasificación.
- Estudio de redes neuronales convolucionales y recurrentes de regresión.

- JdeRobot
- ROS
- Gazebo
- OpenCV
- Python
- PyQt
- Keras
- HDF5



- Circuitos de carreras en Gazebo.
- Piloto autónomo explícito, controlador basado en visión.
- Creación de un conjunto de datos para entrenamiento.
- Piloto autónomo basado en redes neuronales (plantilla).

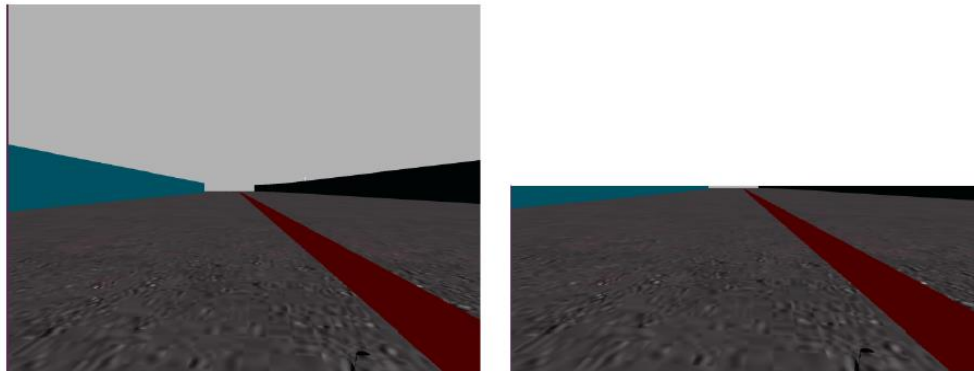




- Cuantificación de velocidades en clases finitas.
- Arquitectura de red: *SmallerVGGNet*:
 - 1 bloque: capa convolucional, BatchNormalization, pooling, dropout.
 - 2 bloques: capa convolucional, BatchNormalization, capa convolucional, BatchNormalization, pooling, dropout.
 - Fully-connected.

Experimentos: imágenes recortadas

- Buenos resultados en métricas neuronales no implican buen rendimiento.



	Programado	5v+7w sesgada recortada		5v+7w sesgada completa	
Circuitos	Tiempo	%	Tiempo	%	Tiempo
pistaSimple (h)	1' 35"	100 %	1' 41"	35 %	
pistaSimple (ah)	1' 33"	100 %	1' 39"	100 %	1' 49"
monacoLine (h)	1' 15"	100 %	1' 20"	100 %	1' 24"
monacoLine (ah)	1' 15"	100 %	1' 18"	100 %	1' 29"
nurburgrinLine (h)	1' 02"	100 %	1' 03"	100 %	1' 10"
nurburgrinLine (ah)	1' 02"	100 %	1' 05"	100 %	1' 07"
curveGP (h)	2' 13"	100 %	2' 06"	95 %	
curveGP (ah)	2' 09"	100 %	2' 11"	7 %	
pista_simple (h)	1' 00"	100 %	1' 02"	8 %	
pista_simple (ah)	59"	100 %	1' 03"	12 %	

- La imagen recortada mejora el rendimiento.

Experimentos: número de clases

- 4 clases de velocidad de tracción (v) y 7 clases de velocidad de rotación (w).
- 4 clases de v y 9 clases de w .
- 5 clases de v y 7 clases de w .
- Gran influencia del número de clases y el rango de clases en el rendimiento.

	Programado	4v+7w sesgada		4v+9w sesgada		5v+7w sesgada	
Circuitos	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo
pistaSimple (h)	1' 35"	100 %	1' 38"	100 %	1' 42"	100 %	1' 41"
pistaSimple (ah)	1' 33"	100 %	1' 38"	100 %	1' 39"	100 %	1' 39"
monacoLine (h)	1' 15"	5 %		5 %		100 %	1' 20"
monacoLine (ah)	1' 15"	5 %		12 %		100 %	1' 18"
nurburgrinLine (h)	1' 02"	8 %		8 %		100 %	1' 03"
nurburgrinLine (ah)	1' 02"	90 %		80 %		100 %	1' 05"
curveGP (h)	2' 13"	100 %	2' 19"	100 %	2' 17"	100 %	2' 06"
curveGP (ah)	2' 09"	100 %	2' 12"	100 %	2' 13"	100 %	2' 11"
pista_simple (h)	1' 00"	100 %	1' 04"	100 %	1' 04"	100 %	1' 02"
pista_simple (ah)	59"	100 %	1' 04"	100 %	1' 02"	100 %	1' 03"

- 5v + 7w es buena combinación.

Experimentos: sesgos en bases de datos

- Conjunto de entrenamiento sin ninguna modificación (red desbalanceada).
- Conjunto de entrenamiento balanceado (red balanceada).
- Entrenamiento con pesos diferentes para cada clase (red sesgada).
- Redes sesgadas mejoran el entrenamiento.

	Programado	5v+7w sesgada		5v+7w balanceada		5v+7w desbalanceada	
Circuitos	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo
pistaSimple (h)	1' 35"	100 %	1' 41"	75 %		100 %	1' 42"
pistaSimple (ah)	1' 33"	100 %	1' 39"	100 %	1' 39"	100 %	1' 43"
monacoLine (h)	1' 15"	100 %	1' 20"	70 %		85 %	
monacoLine (ah)	1' 15"	100 %	1' 18"	8 %		100 %	1' 20"
nurburgrinLine (h)	1' 02"	100 %	1' 03"	100 %	1' 03"	100 %	1' 05"
nurburgrinLine (ah)	1' 02"	100 %	1' 05"	80 %		80 %	
curveGP (h)	2' 13"	100 %	2' 06"	97 %		100 %	2' 15"
curveGP (ah)	2' 09"	100 %	2' 11"	100 %	2' 05"	100 %	2' 15"
pista_simple (h)	1' 00"	100 %	1' 02"	100 %	1' 02"	100 %	1' 01"
pista_simple (ah)	59"	100 %	1' 03"	100 %	1' 03"	100 %	1' 04"

- Redes sesgadas ofrecen mejor rendimiento.

- Predecir valores de velocidades.
- Arquitecturas de red:
 - *PilotNet*: capa de normalización, 5 capas convolucionales, 3 capas fully-connected.
 - *TinyPilotNet*: 2 capas convolucionales, capa dropout, 2 capas fully-connected.
 - *LSTM-TinyPilotnet*: 3 capas convolucionales, capa ConvLSTM2D, 1 capa convolucional, 1 capa fully-connected.
 - *DeepestLSTM-TinyPilotNet*: 3 capas convolucionales, 3 capas ConvLSTM2D, 2 capas fully-connected.

Experimentos: comparativa entre redes

Circuitos	Programado	PilotNet		TinyPilotNet		LSTM-Tinypilotnet		DeepestLSTM-Tinypilotnet	
	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo
pistaSimple (h)	1' 35"	100 %	1' 41"	100 %	1' 39"	100 %	1' 39"	100 %	1' 38"
pistaSimple (ah)	1' 33"	100 %	1' 39"	100 %	1' 38"	100 %	1' 40"	100 %	1' 39"
monacoLine (h)	1' 15"	100 %	1' 21"	100 %	1' 19"	50 %		100 %	1' 22"
monacoLine (ah)	1' 15"	100 %	1' 23"	100 %	1' 20"	12 %		100 %	1' 21"
nurburgrinLine (h)	1' 02"	100 %	1' 03"	100 %	1' 05"	20 %		100 %	1' 05"
nurburgrinLine (ah)	1' 02"	100 %	1' 06"	100 %	1' 06"	80 %		100 %	1' 08"
curveGP (h)	2' 13"	100 %	2' 20"	100 %	2' 11"	100 %	2' 20"	100 %	2' 19"
curveGP (ah)	2' 09"	100 %	2' 16"	100 %	2' 06"	100 %	2' 25"	100 %	2' 18"
pista_simple (h)	1' 00"	100 %	1' 07"	100 %	1' 02"	100 %	1' 11'	100 %	1' 09"
pista_simple (ah)	59"	100 %	1' 09"	100 %	1' 02'	100 %	1' 09"	100 %	1' 08"

- Redes profundas mejoran el resultado.
- Conducción más suave con ConvLSTM2D.

Experimentos: imagen recortada

Circuitos	Programado	PilotNet		TinyPilotNet		LSTM-Tinypilotnet		DeepestLSTM-Tinypilotnet	
	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo
pistaSimple (h)	1' 35"	100 %	1' 37"	100 %	1' 41"	100 %	1' 40"	100 %	1' 36"
pistaSimple (ah)	1' 33"	100 %	1' 38"	100 %	1' 41"	100 %	1' 38"	100 %	1' 37"
monacoLine (h)	1' 15"	100 %	1' 20"	100 %	1' 19"	50 %		100 %	1' 21"
monacoLine (ah)	1' 15"	100 %	1' 19"	100 %	1' 18"	35 %		100 %	1' 19"
nurburgrinLine (h)	1' 02"	100 %	1' 04"	100 %	1' 04"	40 %		100 %	1' 04"
nurburgrinLine (ah)	1' 02"	100 %	1' 06"	100 %	1' 05"	50 %		80 %	
curveGP (h)	2' 13"	100 %	2' 16"	25 %		100 %	2' 17"	100 %	2' 17"
curveGP (ah)	2' 09"	100 %	2' 12"	75 %		100 %	2' 04"	100 %	2' 19"
pista_simple (h)	1' 00"	100 %	1' 04"	100 %	59"	100 %	1' 07"	100 %	1' 05"
pista_simple (ah)	59"	100 %	1' 05"	100 %	1' 00"	100 %	1' 03"	100 %	1' 08"

- La imagen completa mejora el rendimiento.

Experimentos: tipo de imágenes de entrada

- Creación de imagen temporal.
- Imagen apilada (*PilotNet (stacked)*).
- Imagen diferencia (*Temporal (dif)*).
- Imagen apilada-diferencia (*PilotNet (stacked, dif)*).

Circuitos	Programado	PilotNet (stacked)		PilotNet (stacked, dif)		Temporal (dif)	
	Tiempo	%	Tiempo	%	Tiempo	%	Tiempo
pistaSimple (h)	1' 35"	100 %	1' 40"	100 %	1' 43"	35 %	
pistaSimple (ah)	1' 33"	100 %	1' 46"	10 %		10 %	
monacoLine (h)	1' 15"	50 %		5 %		3 %	
monacoLine (ah)	1' 15"	7 %		5 %		3 %	
nurburgrinLine (h)	1' 02"	50 %		8 %		8 %	
nurburgrinLine (ah)	1' 02"	80 %		50 %		3 %	
curveGP (h)	2' 13"	25 %		25 %		12 %	
curveGP (ah)	2' 09"	100 %	2' 07"	75 %		3 %	
pista_simple (h)	1' 00"	100 %	1' 11'	100 %	1' 03"	25 %	
pista_simple (ah)	59"	100 %	1' 08"	100 %	1' 02"	15 %	

- Introducir temporalidad en la red vía imágenes de entrada es complejo

Conclusiones

- **Objetivo cumplido:** Estudio de redes neuronales para conducción autónoma en simulación.
- Aplicación de control visual neuronal.
- Creación de bases de datos propias.
- Estudio de redes neuronales convolucionales de clasificación.
- Estudio de redes neuronales convolucionales y recurrentes de regresión.

Trabajos futuros

- Uso de robots reales.
- Estudio de imágenes apiladas y de imagen diferencia.

Enlaces

- Mediawiki: <https://jderobot.org/Vmartinezf-tfm>
- Repositorio: <https://github.com/RoboticsURJC-students/2017-tfm-vanessa-fernandez>
- Vídeos:
 - <https://www.youtube.com/watch?v=7s4vpMGU2Mg>
 - <https://www.youtube.com/watch?v=J6bDIE7TofE>