Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería Electrónica y Automática Industrial

Trabajo Fin de Grado/Trabajo Fin de Máster

Estudio e implementación de un sistema de detección de la estructura de cruces de carretera utilizando CNNs y Deep Learning

Autor: Adrián Martínez de la Morena

Tutores: Ignacio Parra Alonso

2022

Universidad de Alcalá

Escuela Politécnica Superior

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL

Trabajo Fin de Grado
Estudio e implementación de un sistema de detección de la estructura de cruces de carretera utilizando CNNs y Deep Learning

Autor: Adrián Martínez de la Morena

Tutor: Ignacio Parra Alonso

TRIBUNAL:

Presidente: Dña. Noelia Hernández Parra

Vocal 1º: D. Javier de Pedro Carracedo

Vocal 2º: D. Ignacio Parra Alonso

FECHA: 21/06/2022

Agradecimientos

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

— Douglas Adams

Quería aprovechar esta oportunidad para agradecer a Ignacio y a Noelia, por darme la oportunidad de adentrarme en el mundo del deep learning al que entré sin saber nada y salir sabiendo bastante más, y a la paciencia de Ignacio. Gracias Carlos, por ayudarme a salir del foso en el que me encontré en la recta final y a ti Angelines por insistirme para que no parase. A ti Sara, por mantenerme cuerdo cuando me volvía loco. Ha sido un placer compartir esto con vosotros.

Índice general

Índice de figuras

Índice de tablas

Acrónimos

ADAS	Sistemas Avanzados de Conducción (Advanced Driver Assistance Systems)
ADAM	Adaptative Moment estimation
AI	Artificial Inteligence
AEB	Automatic Emergency Brake
CNNs	Convolutional Neural Networks)
DGT	Direción General de Tráfico
DNN	Deep learning
GPU	Graphical Procesor Unit
INVETT	INtelligent VEhicles and Traffic Technologies
IR	Information Retrival
I+D	Investigación y Desarrollo
KITTI	Karlsruhe Institute of Techno- logy (KIT) and Toyota Technological Institute at Chicago (TTI))
LiDAR	light detection and ranging
LR	Learning Rate
MAP	Mean Average Precision
MAPR	Mean Average Precision@
MBEV	Model-Based Bird Eye View
MMI	Maximun Mutual Information
RAM	Random Access Memory
ResNet	Residual Network
RGB	Red Green Blue image
RNA	Red Neuronal Artificial
RNN	Recurrent Neural Network
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
SGD	Sochastic Gradient Descent
TFG	Trabajo de Fin de Grado
UAH	Universidad de Alcalá de Henares
VPN	Virtual Private Network
2D	Dos Dimensiones
3D	Tres Dimensiones

Resumen

La automatización de los vehículos avanza a pasos agigantados. Más y más, empresas automovilísticas, se están poniendo manos a la obra en el desarrollo de su tecnología para poder competir por la última tecnología del mercado. Muchas han cambiado el equipo de I+D de los motores diésel por el de vehículos eléctricos y su automatización.

Poco a poco los automóviles han ido incorporando sistemas que aumentan la seguridad, así como el confort para la conducción. Gran parte de estos elementos se engloban en el concepto de *Sistemas Avanzados de Conducción (ADAS)*.

El grupo de investigación INVETT está desarrollando un prototipo de vehículo inteligente mediante el empleo del Deep Learning para definir y desarrollar nuevos ADAS, con el objetivo de acercarse al concepto del vehículo autónomo.

Este TFG se integra junto con el resto de trabajos del grupo de investigación INVETT para la implementación de un sistema de detección de la estructura de cruces de carretera utilizando CNNs y Deep Learning.

El alcance concreto de este TFG es la generación de una base de datos de imágenes MBEV mediante nubes de puntos creadas por un LiDAR tras lo cual se usará esta base de datos para la realización de diversos entrenamientos de redes neuronales con el fin de que éstas, tras ser entrenadas, consigan discriminar entre siete intersecciones diferentes así como su análisis y posterior interpretación.

Para estos entrenamientos se hace uso de varias bases de datos, obtenidas de otros estudios (KITTI y KITTI-360) y una base de datos complementaria que se ha creado con el vehículo del grupo de investigación de la Universidad de Alcalá, INVETT.

Palabras clave: Aprendizaje profundo, Redes Neuronales Artificiales, Conducción Autónoma, Segmentación Semántica, Clasificación.

Abstract

Automation in vehicles is growing fast now a days, almost all companies are investing in this topic. Some of it has changed the investigation of the Diesel engines to automation and AI investigation.

Slowly, cars are implementing different systems that improve security and confortability for the driver. All this systems are gather into one big group called *Advanced driver-assistance systems (ADAS)*

This thesis cooperates with the investigation group *INVETT* in the implementation of intersection detection system using deep learning and CNNs. The aim is to create an MBEV image dataset from a LIDAR point cloudto make trainings of neuronal networks in order to get them to distinguish all seven types of intersections. TO do so, the trainings are done with several databases (KITTI, KITTI-360 and a small database created by UAH)

Capítulo 1

Introducción

1.1 Motivación

Hoy en día, en la mayoría de las ciudades hay una creciente densidad de población, por tanto de vehículos en circulación y, como consecuencia directa, una alta siniestralidad, debido a que se presentan una cantidad muy alta de estímulos que causa un aumento considerable del riesgo.

Como se puede observar en la figura 1.1, los accidentes más usuales son los accidentes de coches sin que haya otro elemento implicado. En segundo puesto, se encuentran los accidentes causados por coches a las personas, seguidos por los accidentes de coches contra coches.

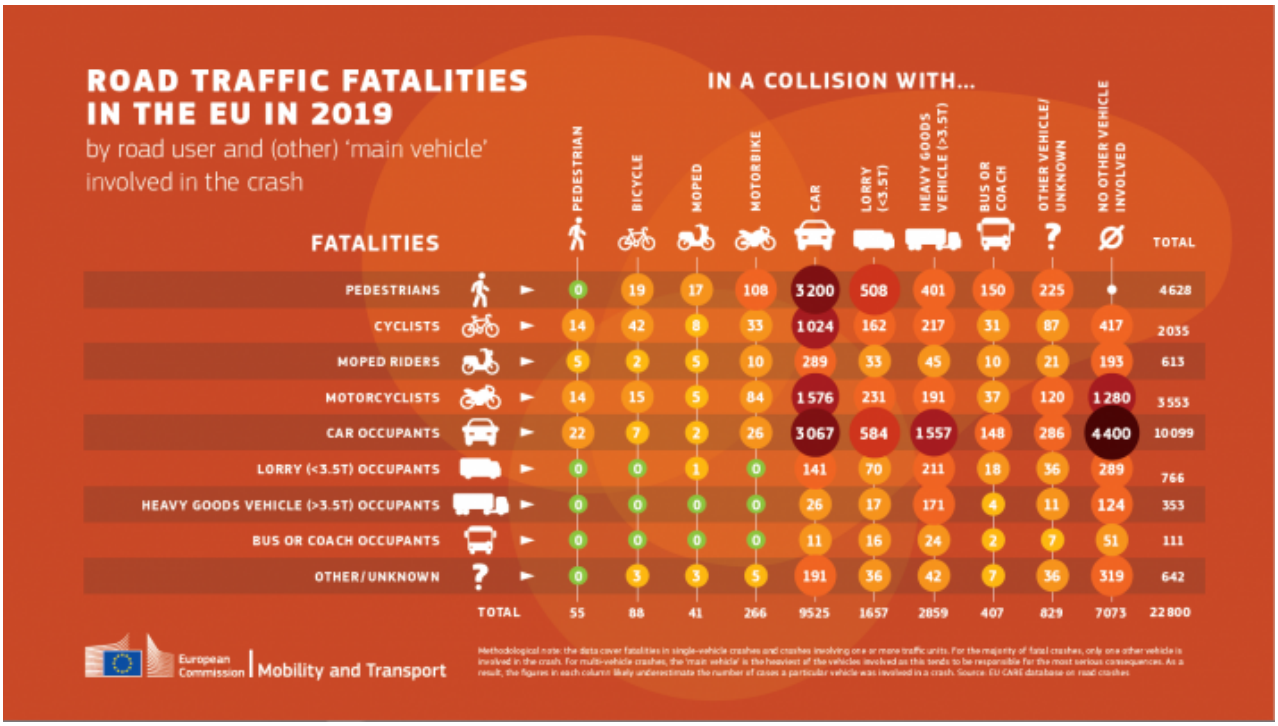


Figura 1.1: Accidentes entre los principales usuarios [2]

En la gráfica 1.2, se puede observar que, tras las colisiones con vehículos parados, los golpes en intersecciones laterales son los más predominantes en las ciudades.

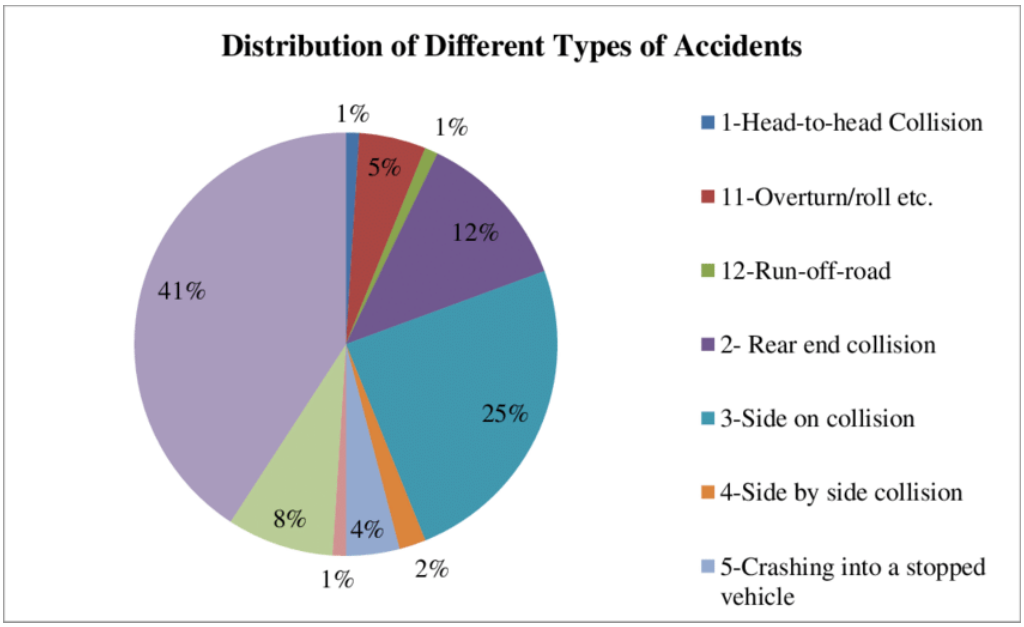


Figura 1.2: Accidentes en diferentes intersecciones [3]

Según indica la *Dirección General de Tráfico (DGT)*, en las carreteras nacionales de España, las intersecciones representan el 40 por ciento de los accidentes [4].

Por ello es importante que los sistemas autónomos sepan interpretar de manera segura estas situaciones y, en concreto, la interpretación de intersecciones ya que, en estás, se concentra el mayor número de accidentes. Para ello, la detección y el entendimiento en definitiva de estos escenarios es muy importante tanto para los vehículos autónomos como para los ADAS en general, con el objetivo de prevenir los accidentes de tráfico, así como aumentar la seguridad de los vehículos más vulnerables.

La tecnología actual respecto a la creación y disponibilidad los sistemas de interpretación de carreteras, está lejos de alcanzar el nivel de autonomía integral, especialmente en zonas muy concurridas como son los intersecciones. Por ello, la circulación por estas zonas, requiere sistemas que sepan interpretar e identificarlas adecuadamente. Por ello es necesario reforzar las capacidades de los ADAS lo cual se ve posible mediante la aplicación de las tecnologías basadas en redes neuronales.

A priori, el gran número de estímulos que representan las intersecciones para un conductor, no supondrá para los sistemas inteligentes más que una serie adicional de variables a tener en cuenta.

Como se observa en la figura 1.3 a muy alto nivel, la idea es que el vehículo interaccione con el resto de variables externas para poder predecir y adelantarse a posibles accidentes, puesto que el número de variables y las reacciones dependen de la clase de intersección que se trate. Así, en la 2.4 se pueden ver las diferentes clases de intersecciones que se consideran relevantes para la gestión inteligente de intersecciones.

El primer, por tanto, paso para el procesado de intersecciones por un sistema inteligente es la clasificación de las mismas, ya que según sean, se tratarían de diferente forma.

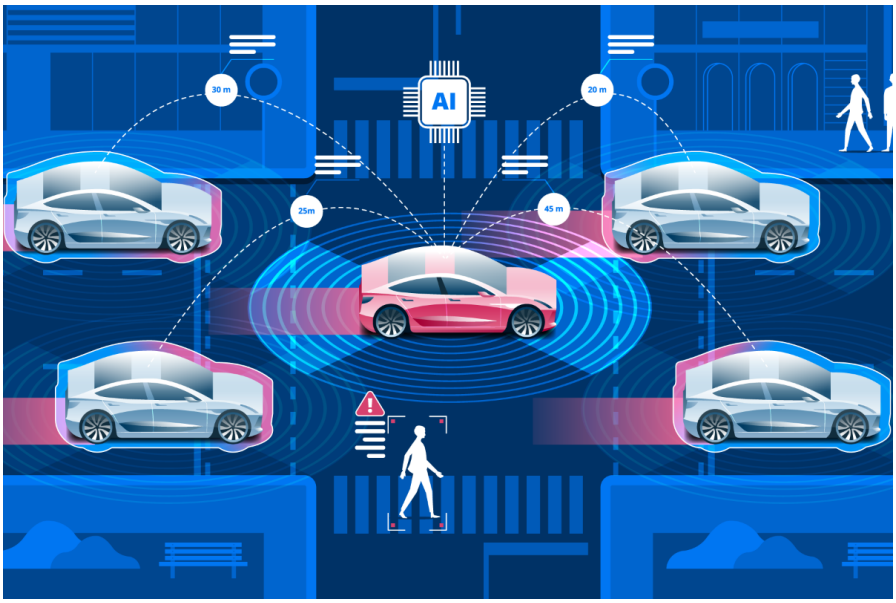


Figura 1.3: Gestión de variables de una intersección mediante AI [1][5]

Hasta ahora, como método habitual para el procesamiento de imágenes, se emplean las imágenes obtenidas por cámaras RGB. Este método, como analizaremos más tarde, implica registrar una gran cantidad de ruido junto con los datos netos de imagen que perturba el procesado de las mismas. Igualmente veremos, como método alternativo al procesamiento de las imágenes con cámaras, la aplicación de los sistemas *LIDAR* que generan nubes de puntos a partir de la información captada, la cual se empleará posteriormente para el entrenamiento de las redes neuronales.

[1] sacado de https://blogs.iadb.org/transporte/wp-content/uploads/sites/9/2020/01/vehiculos-autonomos-1.jpg/

1.2 Objetivos

Este trabajo tiene por objetivo conseguir que el vehículo aprenda a distinguir las diferentes intersecciones de carretera, y así, reducir la ambigüedad de las situaciones de localizaciones falsas que pueden surgir en escenarios típicamente urbanos.

Para ello, a lo largo del trabajo, se estudiarán varias de las arquitecturas actuales de CNNs, para entender su funcionamiento, y seleccionar la más apropiada para su posterior implementación y aplicación a un caso práctico de la gestión de intersecciones.

En definitiva, con la realización de este Trabajo Fin de Grado se persigue la consecución de los siguientes objetivos:

- Comprender los principios de funcionamiento de los distintos tipos de Redes Neuronales.
- Empleo de redes convolucionales en su aplicación a problemas de detección de objetos en imágenes.
- Evaluar si el uso de nubes de puntos de imágenes LIDAR es una alternativa al empleo de imágenes RGB como base de datos para el entrenamiento de redes neuronales.
- Conocer distintas tipologías y configuraciones de Redes Neuronales Convolucionales.
- Aprender a implementar una Red Neuronal Convolucional con sus hiper-parámetros de ajuste y en la plataforma elegida.
- Analizar los resultados obtenidos y plantear posibles continuaciones.

Por otro lado, con la realización de este Trabajo Fin de Grado, se contempla cubrir los objetivos generales de este tipo de trabajos:

- Búsqueda bibliográfica de información relativa al campo de estudio.
- Estudio y comprensión de las distintas clases de CNNs.
- Realización de pruebas para resolver el problema planteado.
- Redacción de una memoria que presente y exponga el trabajo realizado, los objetivos alcanzados y los resultados obtenidos.
- Aprendizaje y empleo de Python como lenguaje de desarrollo software.
- Aprendizaje y empleo de Latex como editor avanzado de textos.

1.3 Organización de la memoria

La estructura del TFG consta de seis partes:

- En el capítulo 1. Introducción, se plantean los aspectos que han motivado la selección del tema para este TFG seguido del planteamiento de los objetivos que se persiguen en el mismo.
- En el capítulo 2. Estado del arte, se revisan las diferentes fuentes de información (bibliografías, investigaciones, estudios, etc.) en la aplicación al procesamiento de imágenes y discriminación de intersecciones en tiempo real, así como las diferentes redes neuronales y los hiper-parámetros relevantes en su entrenamiento.
- En el capítulo 3. Descripción del sistema. Donde se explican los dispositivos y recursos usados para la creación de las diversas bases de datos para el posterior entrenamientos de la red neuronal del sistema.
- en el capítulo 4, se explica el planteamiento de las pruebas así como el método de obtención de los datos.
- En el capitulo 5, se comentan los resultados obtenidos, las razones posibles por las que se han obtenido esos datos, así como, las posibles modificaciones para la mejora de estos mismos.
- En el capitulo 6, se explican las conclusiones que se han obtenido de todo el trabajo así como las posibles lineas futuras.

Capítulo 2

Base teórica

2.1 Estado del arte

Con el crecimiento de los ADAS, más y más aspectos de la conducción han sido transferidos del humano a los sistemas de control del propio vehículo. Para poder manejar todo esto, los vehículos deben poder saber interpretar tanto diversas situaciones como sus alrededores, y así poder ajustar su comportamiento acorde con esto.

Una muestra de algunos de los ADAS se ilustran en la figura 2.1

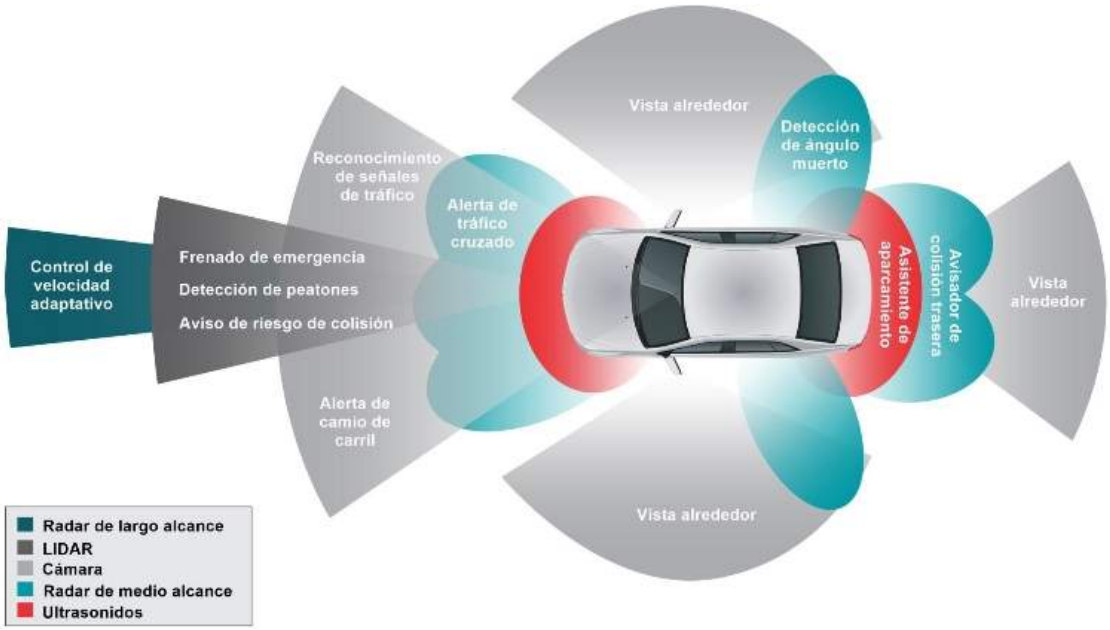


Figura 2.1: Esquema gráfico de sistemas ADAS [6] [1]

Ya en 1917, la doctora June Mccarroll inició el interés por mejorar la circulación de las carreteras incorporando las marcas viales [7]. Esto suscitó las investigaciones sobre la mejora de las carreteras y en concreto sobre la automatización de la conducción. En 1987, Todd R Kushner y Sunil Puri desarrollaron los primeros sistemas de detección de intersecciones a través de la coincidencia de imágenes de una base de datos de mapas.

[1] sacado de `https://biriska.com/conoces-los-modernos-sistemas-adas-de-los-coches/`

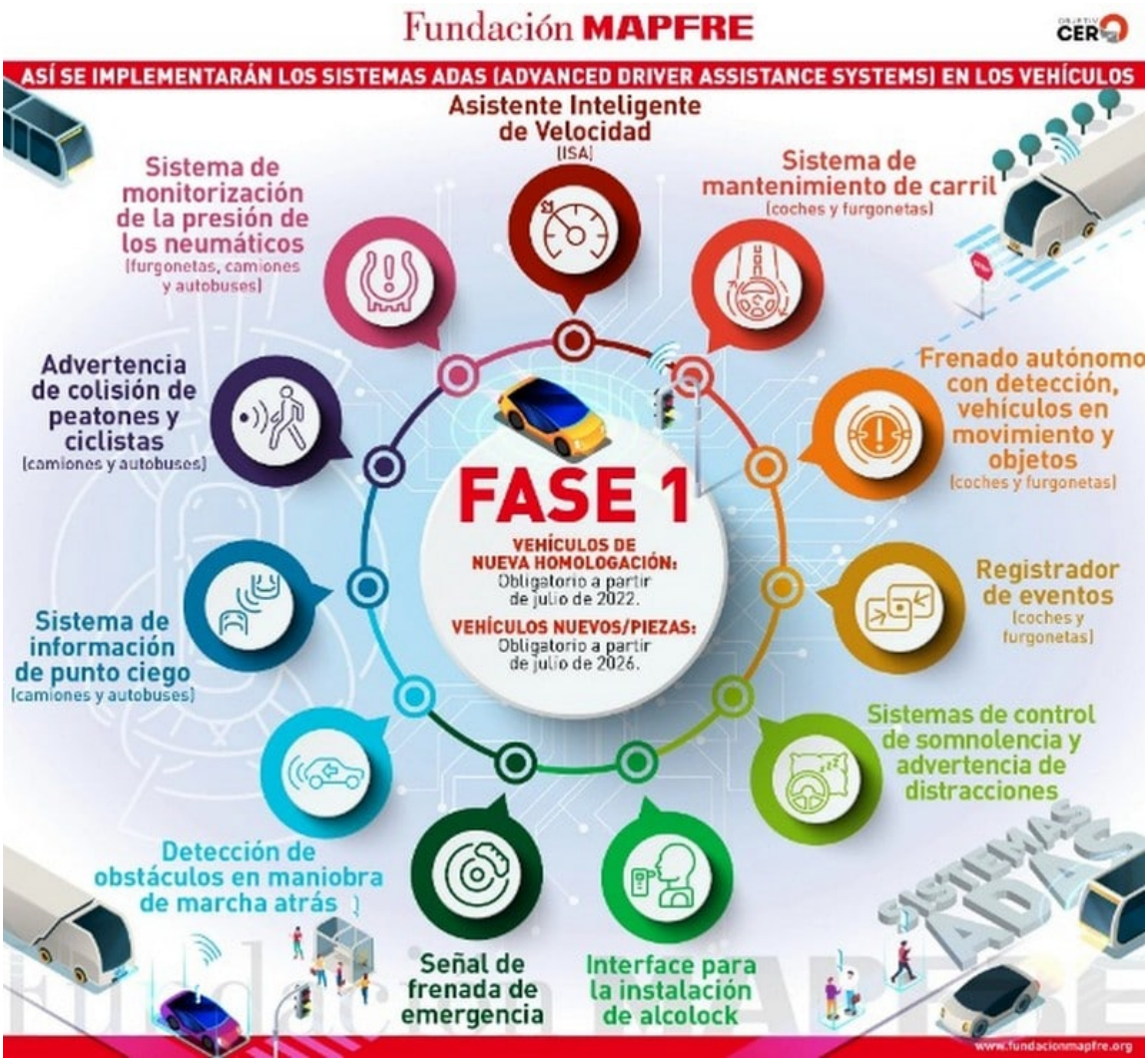


Figura 2.2: Diferentes ADAS [2][8]

Pocos años después, en 2012, los coches empezaron a incorporar *ADAS* tales como 2.2:

- Detección de salida de carril involuntaria, la cual funciona con la ayuda de unos sensores infrarrojos situados en la parte inferior del coche que detectan las lineas del carril y a partir de una velocidad determinada en la centralita, activan el control del volante, evitando la salida de la carretera por distracción.
- Automatización de las luces de larga distancia, que se activan y desactivan si detectan luces acercándose al vehículo.
- Detección de fatiga del conductor, esta funciona con un sensor de ángulo de giro del volante, que analiza los movimientos del conductor junto con un contador que analiza el tiempo que lleva el vehículo en movimiento y, si los movimientos del volante difieren mucho del comportamiento habitual y lleva más de un tiempo determinado conduciendo, avisa al conductor para que pare a descansar.

[2]sacado de `https://www.circulaseguro.com/sistemas-adas-web-mapfre/`

- El detector de ángulo muerto, que detecta elementos ubicados en éste, avisando al conductor por medio de un indicador luminoso en el retrovisor.
- *AEB (Autonomous Emergency braking)*, el cual mediante un sensor LASER detecta obstáculos en la trayectoria del vehículo y si hay peligro de colisión, activa automáticamente los frenos.
- *AEB de peatón*, que funciona de manera similar al sistema anterior pero velocidades inferiores a 60 km/h
- Adaptación automática de luces, que modifica el haz de luz según la posición del vehículo en el sentido contrario.
- Adaptación activa de la amortiguación, la cual según el terreno por el que se circula, modifica la dureza de la misma.

Estos, entre otros, son sistemas de *ADAS* que han ayudado en los últimos años a la reducción de accidentes relacionados con el conductor. Pero, por el momento, no incorporan la interpretación o diferenciación del tipo de intersección.

Grandes compañías como Tesla, Toyota, VolskWagen, Google, Jaguar, Mercedes Benz, etc., están avanzando en la automatización de los vehículos. En función de la dificultad que supone su implantación en cuanto a la adaptación tecnológica, de las carreteras y de los usuarios, se definen, como se muestra en la figura 2.3, seis niveles de automatización de vehículos:

- **Nivel 0**, sin automatización en la conducción: en este nivel, el conductor tiene completo control del vehículo.
- **Nivel 1**, asistencia en la conducción: en este nivel, se incorporan algunas de las ayudas anteriormente mencionadas, que solo facilitan la conducción pero dejando el control del vehículo en manos del conductor.
- **Nivel 2**, autonomía parcial: en este nivel, se requiere del conductor aunque el mismo sólo supervisa y podría tomar el control en caso de necesidad.
- **Nivel 3**, autonomía condicionada: este nivel tiene una autonomía mayor a la anterior pero, se precisa del conductor para que actúe cuando el vehículo lo exija.
- **Nivel 4**, automatización elevada: este nivel no precisará de la atención del conductor.
- **Nivel 5**, automatización completa: el vehículo podrá prescindir de elementos de conducción como volante o pedales.

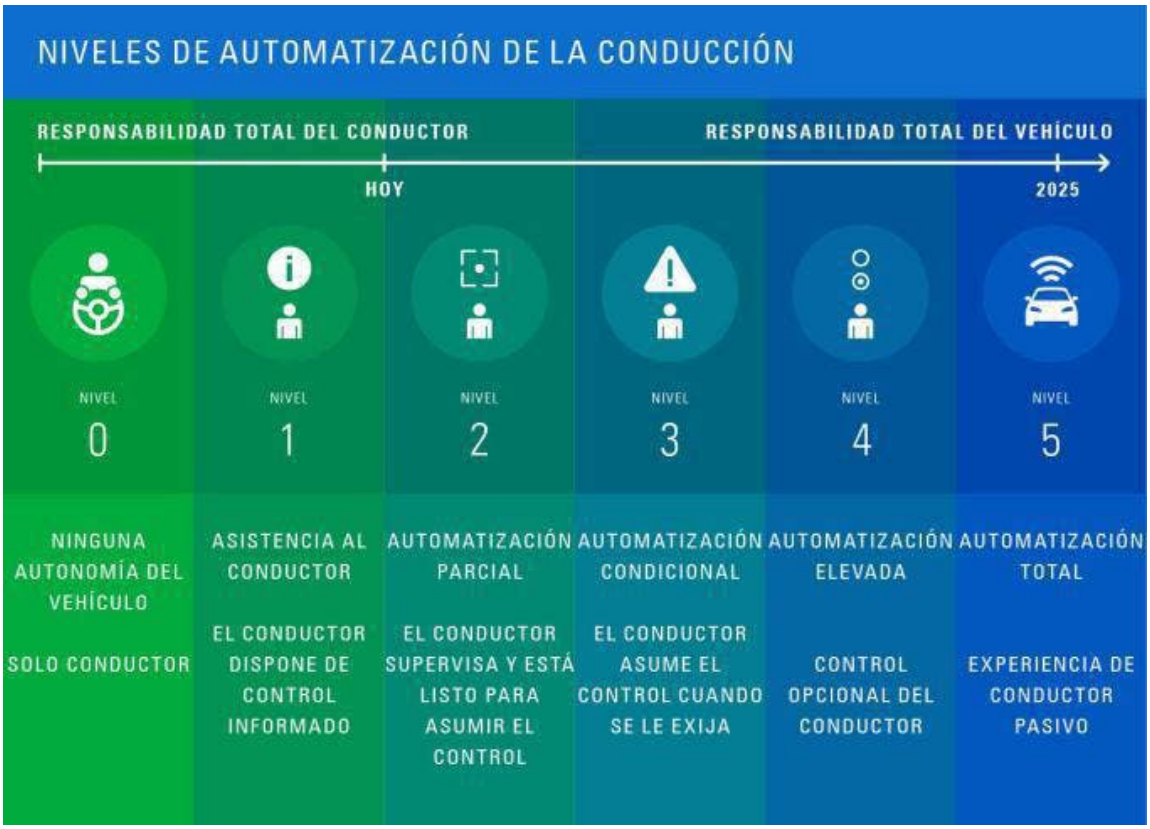


Figura 2.3: Niveles de conducción autónoma [3][9]

2.1.1 Tipos de intersecciones

Para la automatización de la conducción, los ADAS deben discriminar los diferentes escenarios a los que se enfrenta el vehículo. Como se ha comentado anteriormente, uno de los escenarios que representa mayor riesgo son las intersecciones. Nos centraremos en este aspecto a partir de ahora en este TFG.

Se define como intersección la zona en la que confluyen dos o más vías. Los tramos de carreteras que confluyen en la intersección se denominan ramales. Las intersecciones constituyen una parte esencial de la red viaria, ya que son los puntos en los que se puede cambiar de vía para seguir la ruta deseada. En ellas los vehículos pueden seguir distintas trayectorias, y es necesario ordenarlas para reducir los conflictos entre los distintos movimientos.

La detección de futuras intersecciones, puede ayudar a mejorar varios aspectos en el contexto de la conducción autónoma, así como, la predicción de otros usuarios o el ajuste del sistema con respecto a los diferentes tipos de escenarios.

Uno de los aspectos a tener en cuenta para el correcto comportamiento de los vehículos es el segmento de carretera inmediatamente delante del mismo, así como la presencia y el tipo de intersección. Esto define el escenario y proporciona información relevante para la conducción.

[3]sacado de https://www.hibridosyelectricos.com/media/hibridos/images/2021/03/11/20210311114044165 21.jpg/

Los autores en [10], exploran imágenes *RGB* tomadas de cámaras situadas en la parte frontal que, posteriormente, se procesan para crear una red temporal de ocupación, y que se usan para comparar formas predeterminadas y analizar ocupación en la siguiente intersección. Diferente enfoque puede hacerse en investigaciones en las que se hace uso del aprendizaje profundo para distinguir intersecciones [11].

Desde un punto de vista técnico, aparte de este enfoque empleado para analizar las imágenes de cámaras, existen algoritmos que se basan en *LIDAR* o, incluso, la combinación de la información de las imágenes como de los sensores.

Un enfoque similar se consigue en la investigación de [12], en la que los autores sugieren el uso de dos paquetes de imágenes relativas a una intersección procesada con *DNN y RNN*.

Respecto al uso de sensores *LiDAR*, existen varios puntos de vista. En esta investigación [13], los autores hacen uso de *LiDAR* y de entrenamientos de aprendizaje profundo (deep learning) para entrenar mediante la información recibida por los sensores.

Como hemos mencionado más arriba, las intersecciones son críticas desde el punto de vista de la seguridad, tanto en zonas urbanas e interurbanas. Así, en España, en el año 2009, el 48 por ciento de los accidentes con víctimas se localizaron en cruces, y en éstos, fallecieron 210 personas, lo que supone el 36 por ciento del total de muertos en zona urbana, y resultaron heridas 29.907 personas. En carretera, teniendo en cuenta el menor número de intersecciones, este porcentaje es también muy elevado, alcanzando el orden del 20 por ciento.

Acorde con investigaciones anteriores [14] [1], se propone diferenciar entre 7 tipos de intersecciones ilustradas en la imagen 2.4. Aun con la limitación de las siete clases, esto será un avance respecto al estado del arte actual.

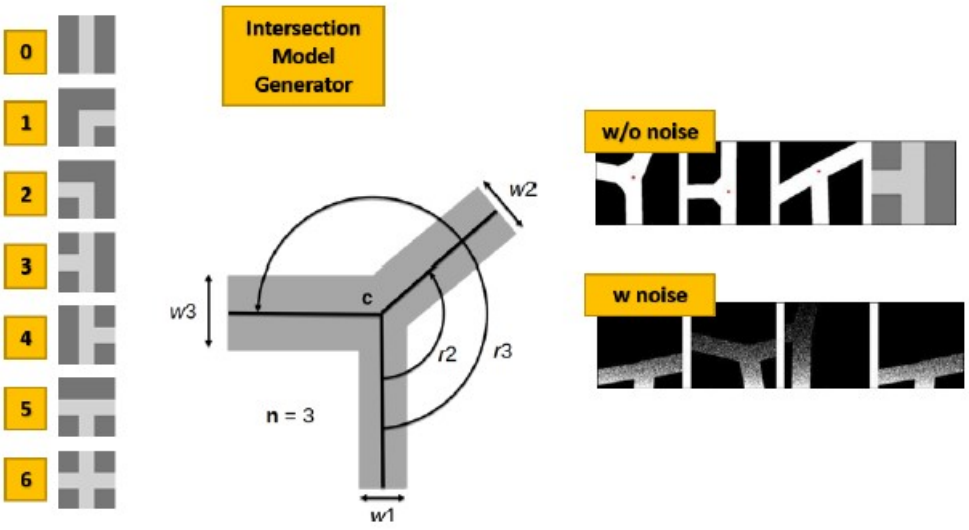


Figura 2.4: Intersection model generator [1]

El generador de intersecciones simples junto con los siete tipo de intersecciones se visualiza en la imagen 2.4. Los parámetros de que constan estos tipos de intersecciones, posibilita, no solo la tipología de la intersección o el número de calles que la componen, sino la anchura de cada calle y la

localización del centro con respecto a la imagen. Este modelo, permite generar imágenes *BEV* que contienen todas las diferentes tipologías de intersecciones que puede ser encontradas en los paquetes de datos que serán usados durante las fases de entrenamiento.

Para tener más exactitud a la hora de analizar las intersecciones se utiliza un estimador de superficies de suelo y un segmentado de nubes de puntos de carretera, que se explicará más adelante, que estiman la elevación de la superficie del suelo en tiempo real.

2.1.2 Redes neuronales

Las redes neuronales han ido alcanzando cada vez un mayor auge, teniendo una gran variedad de usos en un amplio rango de campos y dando soluciones sencillas a problemas, cuya resolución resultaría compleja si se hiciese de otra manera. Una *Red Neuronal Artificial (RNA)*, es un sistema de procesamiento de información, que tiene ciertas características de funcionamiento en común con las redes neuronales biológicas.

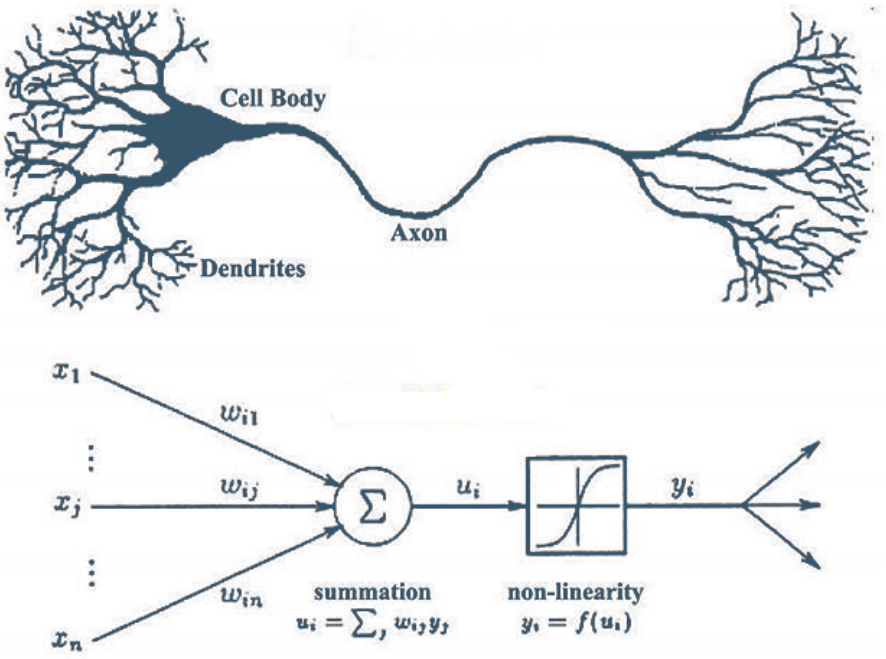


Figura 2.5: comparación de neurona con red neuronal[4][15]

Las *RNAs* se han desarrollado como generalizaciones de modelos matemáticos del conocimiento humano. Existen varios tipos de redes neuronales:

- **Red neuronal convolucional (CNN)**: Se inspiran en la visión de los animales. Se usan en aplicaciones como procesado de imágenes, reconocimiento de vídeo y procesado de lenguaje natural. Una convolución es una operación matemática donde una función es aplicada a otra función y da un resultado de una mezcla de las dos. Las convoluciones son buenas para detectar estructuras simples en una imagen y juntar estas características para crear otras más complejas.

  El CNN es un algoritmo de *Deep learning* que esta diseñado para trabajar con imágenes como entradas. A éstas se les asigna una etiqueta para poder diferenciarlas entre sí. Estas redes se asocian comúnmente con imágenes como datos de entrada, pero se pueden usar para otro tipo de datos de entrada.

- **Fully Convolutional Neuronal Networks (FCNNs)**: Propone un método basado en un codificador-decodificador con predicción *pipeline* para segmentar los píxeles de una imagen en cada una de sus clases. La parte del codificador se encarga de muestrear las imágenes y

[4]sacado de `https://magiquo.com/redes-neuronales-o-el-arte-de-imitar-el-cerebro-humano//`

sintetizarlas en un paquete de en características relevantes. Tras esto, la capa de clasificación del codificador se descarta y el paquete de características relevantes se ceden al decodificador. Éste aplica una conversión de up-sampling a través de las capas deconvolucionales, hasta que se obtiene mapa de probabilidad del mismo tamaño que el de la imagen de entrada para cada una de las clases. Los parámetros de las capas deconvolucionales pueden ser aprendidos durante procesos de aprendizaje de la misma manera que lo hacen las capas convolucionales.

En la figura 2.6 representa un proceso de encoder-decoder, en el que, tras pasar la información por este proceso, extrae las características de las imágenes en este caso.

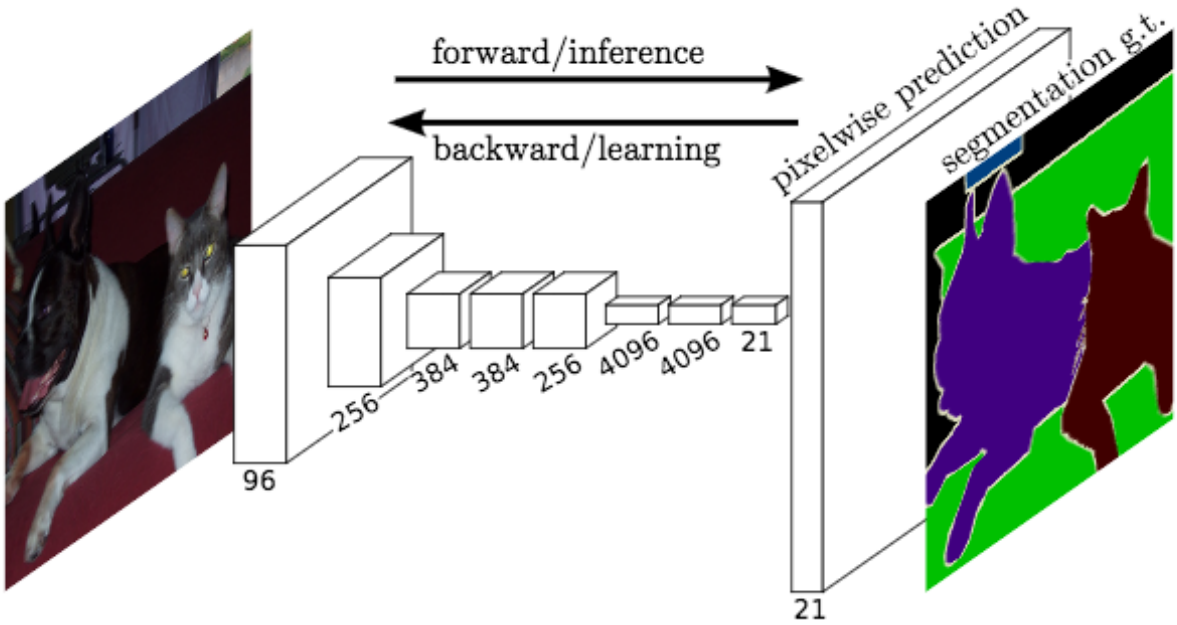


Figura 2.6: fully convolutional networks[5][16]

- **Restricted Deformable Convolution:** Una versión de las convoluciones deformables introducido en [17]. Estas particulares convoluciones están basadas en corregir la falta de información espacial que las CNNs tienen.
- **Redes neuronales no recurrentes:** Estas redes trabajan, a diferencia de las demás, en un solo sentido sin realimentación y carecen de memoria. Estas redes son las menos usadas hoy en día.
- **Redes neuronales recurrentes:** En estas, las neuronas tienen la posibilidad de realizar conexiones de realimentación ya sea entre neuronas de una misma capa o entre diferentes capas.

El entrenamiento *Teacher-Student* se trata de una transferencia de conocimientos, donde, como se explica anteriormente, una red *Teacher* se entrena con un paquete de datos simplificado y después, es usada para enseñar a una red *Student* que ha adquirido los conocimientos de la red anterior y mediante un nuevo entrenamiento con una base de datos mas compleja, se espera que consiga los mismos resultados. Inicialmente, esa técnica se creó para la formulación de modelos de compresión, pero

[5]sacado de `https://production-media.paperswithcode.com/methods/new_alex-model.jpg//`

también ha sido usado para adaptación de dominios como se ilustra en la figura 2.7 El entrenamiento *Teacher-Student* tradicionalmente se usa para compresión de modelos como en [18]. También se ha aplicado para adaptación del dominio [19], donde la red maestra, es entrenada en el dominio fuente y la red *Student* es entrenada en el dominio objetivo. Es particularmente efectivo cuando bases de datos paralelas están disponibles en fuente y objetivo. En el estudio [20], una gran cantidad de datos no supervisados paralelos se usa para mejorar el rendimiento de los modelos en el dominio objetivo.

Desde un punto de vista técnico, la idea detrás de los entrenamientos *Teacher/student*, incluye la transferencia de conocimiento entre un dominio mas simple a uno mucho mas complejo. En esta investigación, el dominio simple del que proponemos realizar el aprendizaje, consiste en un conjunto de imágenes de *vista de ojo de pájaro (BEV)* que son usadas en [1], para la clasificación de intersecciones y localización de vehículos.

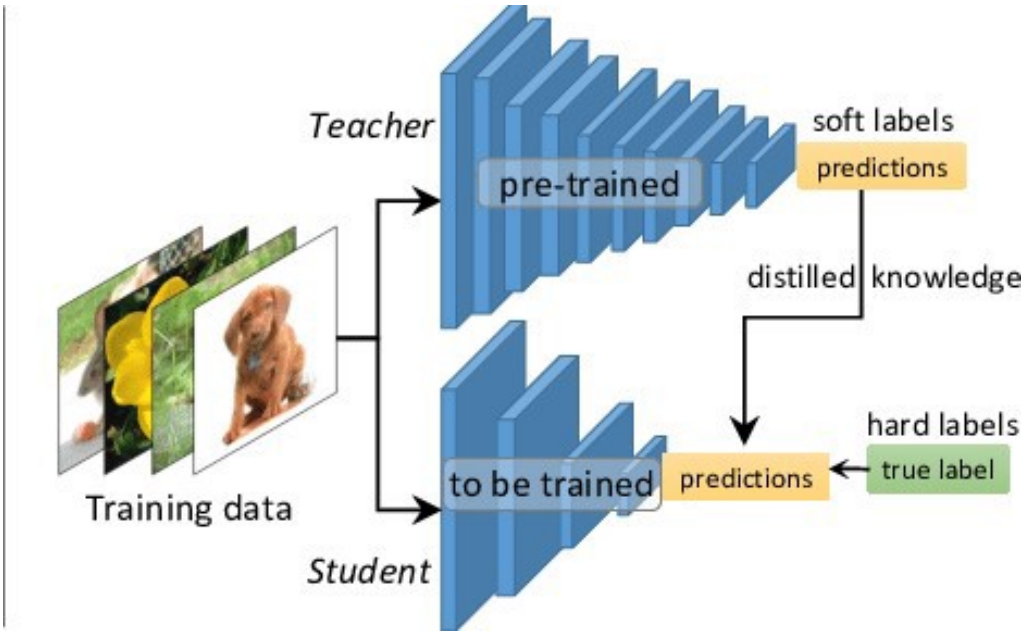


Figura 2.7: Teacher student [6][21]

La transferencia de conocimiento es el método general de *Machine learning* para transferir el conocimiento de un modelo a otro. Dependiendo del contexto, tiene un nombre u otro. En este caso, donde se tiene que aprender un pequeño modelo en el mismo dominio, el método se llama: *adaptación del modelo*. En el caso que se tenga que aprender un modelo en diferentes dominios, se llamará *adaptación del dominio*. La transferencia de aprendizaje ha sido aplicada para acelerar el proceso en varios ajustes. El artículo [22] ofrece una buena comparación de métodos para acelerar el proceso.

Así como este tipo de entrenamiento, existen infinidad de otros entrenamientos para cada caso especifico. Este es el caso del *Cyclical training*, definido como un conjunto de ajustes con los que el entrenamiento empieza y termina con *easy training*. El *hard training* ocurre durante las iteraciones intermedias, es decir, puede ser considerado como una combinación de *curriculum learning* en las primeras iteraciones, e iteraciones con *fine-tuning* al final del entrenamiento junto con un entrenamiento sobre el problema completo para una mejor generalización. Se ha demostrado que en muchos

[6]sacado de `https://github.com/ShivamRajSharma/Teacher-Student-Network//`

aspectos el aprendizaje de redes neuronales aparecen en las primeras iteraciones del entrenamiento [23].

2.1.3 Entrenamiento de Redes neuronales

A la hora de entrenar una red, hay que tener en cuenta varios parámetros a modelar con el objetivo de ajustar el entrenamiento para conseguir los mejores resultados posibles.

- ***Learning Rate(LR)***: Es un hiper-parámetro de entrenamiento que controla cuánto se ajustan los pesos de nuestra red neuronal con respecto al gradiente de pérdidas. Los límites del LR oscilan entre 0 y 1, y según los resultados obtenidos, que se analizan con las gráficas obtenidas, se habrá de aumentar o disminuir para mejorar la efectividad del entrenamiento. Los entrenamientos de redes neuronales usan algoritmos de descenso del gradiente, esto es una optimización que estima el error del gradiente del estado actual usando ejemplos de paquetes de datos, y después, actualiza los pesos del modelo usando *"back propagation"*. La cantidad que los pesos son actualizados durante este proceso son referidos como *LR*. Es decir, el *LR* controla cuan de rápido se adapta el modelo al problema. Valores pequeños requieren más *epochs* dado que los cambios son pequeños entre cada actualización de los pesos. Valores altos en cambio, requieren un menor número de actualizaciones.

  El problema viene al elegir el valor del *LR*, ya que, si el valor es demasiado alto para el problema, resultará en una convergencia muy alta siendo no óptima la solución al problema, así como, si el valor es demasiado pequeño causará que el proceso se bloquee.

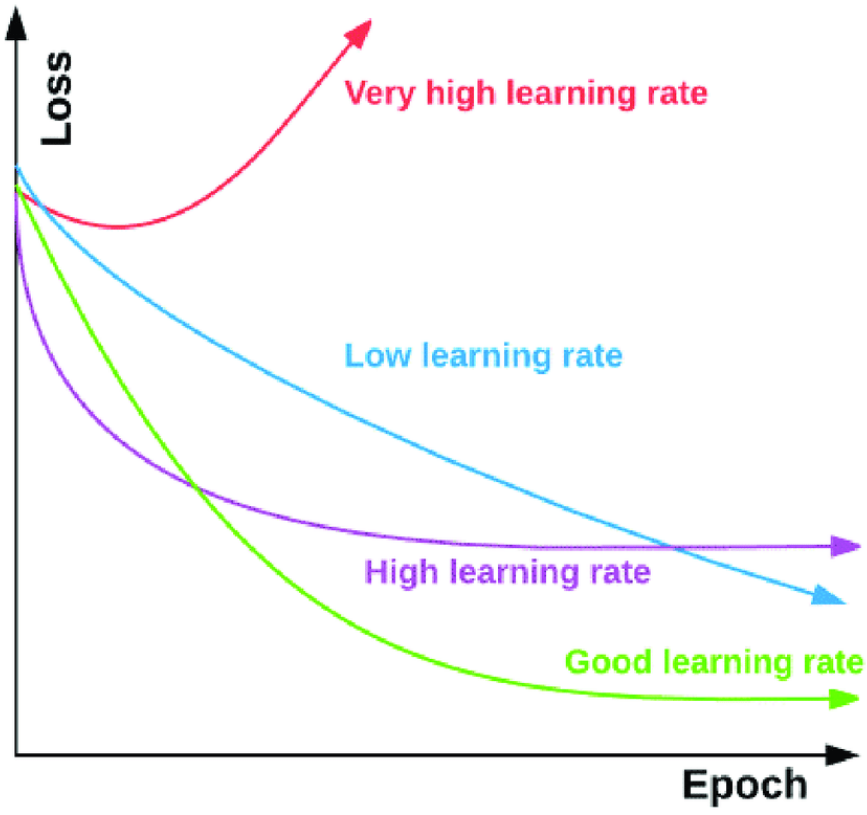


Figura 2.8: Learning rate [7][24]

[7]sacado de https://www.researchgate.net/figure/Changes-in-the-loss-function-vs-the-epoch-by-the-lea
fig2_341609757//

- ***Optimizador***: Durante el proceso de entrenamiento, los valores de los pesos se van cambiando con el objetivo de minimizar la función de pérdidas y hacer las predicciones lo más correctas posibles. Aquí es donde aparece el *optimizador*. Éste enlaza la función de pérdidas junto con los parámetros del modelo actualizando el modelo, es decir, el optimizador modela y da forma al modelo de la manera más precisa posible modificando los pesos de la función de pérdidas del optimizador indicando si se está moviendo en la dirección correcta.

  Existen varios optimizadores, entre otros, hemos probado con:

  1. **Stochastic Gradient Descent (SGD)**: Este optimizador limita el cálculo de la derivada a tan solo una observación por batch.
  2. **Root Mean Square Propagation (RMSprop)**: Este optimizador mantiene el factor de entrenamiento diferente para cada dimensión, pero el escalado del factor de entrenamiento se realiza dividiéndolo por la media del declive exponencial del cuadrado de los gradientes.
  3. **Adaptative moment estimation (ADAM)**: En este algoritmo, se mantiene un factor de entrenamiento por parámetro y además de calcular *RMSProp*, cada factor de entrenamiento también se ve afectado por la media del momento del gradiente.

- ***Arquitecturas de red:*** Las arquitecturas de red intentan reproducir el comportamiento del cerebro del ser humano y así, resolver problemas complejos.

  Se denomina arquitectura de red a la topología, estructura o patrón de conexión de una red neuronal. En una red neuronal los nodos se conectan por medio de sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. La información en estas conexiones solo se puede desplazar en un solo sentido. En general las neuronas se agrupan en unidades neuronales denominadas *capas* el conjunto de estas capas se denomina red neuronal.

  Una arquitectura de red está creada por unidades individuales que se llaman neuronas que simulan el comportamiento de las neuronas del cerebro. Define la forma en que se estructura un modelo de aprendizaje profundo. La arquitectura determina la precisión del modelo (entre otros factores que afectan a su precisión).

  Existen muchos tipos de arquitecturas de redes *CNN*:

  - **Inception**: Es una red neuronal no convolucional con un diseño que consiste en repetir componentes referidos a los módulos de inicio. Este nombre se le da debido a la película de Christopher Nolan 'Inception'
  - **ResNet (Residual Network)**: Es un tipo de red neuronal que replica la estructura piramidal de las células cerebrales.
  - **MobileNet**: Este tipo de red neuronal usa convoluciones profundas separables en vez de las convoluciones normales con el objetivo de reducir el tamaño del modelo así como su computación, por ello puede ser útil para crear redes neuronales ligeras para móviles.

[8] sacado de `https://miro.medium.com/max/1320/1*A0JzlOwTokGwhcBhT89tDQ.png//`

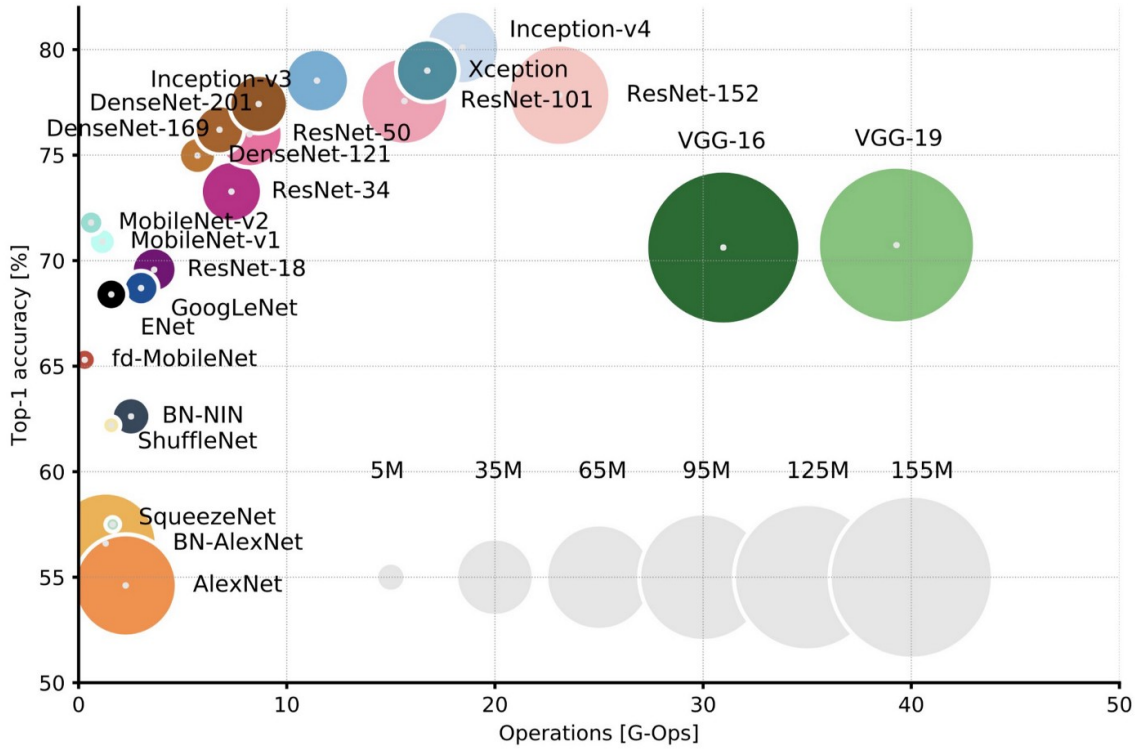


Figura 2.9: arquitecturas de red[8][25]

En la figura 2.9 se pueden observan las diferentes arquitecturas de red que se usan en la actualidad clasificadas según su exactitud respecto al número de operaciones realizadas, Se observa que hay varios modelos de redes: Inception, ResNet, MobileNet...

Otros ajustes a tener en cuenta a la hora de entrenar:

- **Early stopping**: Cuando se entrenan redes neuronales, se llega a un punto del entrenamiento en el que se deja de generalizar y se empieza a memorizar. Cuando ocurre esto, la red neuronal deja de aprender y el *Learning rate* disminuye notablemente. Por ello, es importante saber cuándo se debe dejar de entrenar para no romper con lo aprendido, aquí es donde aparece el *early stopping*. Esto requiere que la red se configure con más capacidad de la que requiere el problema. Cuando se entrena una red neuronal, un mayor número de *epochs* de entrenamiento se usan de lo que normalmente se usarían. Una vez que se elije el modelo de evaluación, habrá que elegir un proceso que lance el *early stopping*. Usará un monitor de rendimiento de la red.
- **Patience**: este valor informa de cuantas *epochs* debe continuar cuando detecta un descenso del rendimiento.
- **Patience start**: son elementos que paran el entrenamiento si la curva de aprendizaje se traba en un valor constante, lo cual quiere decir que está dejando de aprender y empieza a memorizar.
- **Métricas de aprendizaje**: Esto intenta mapear datos al espacio de clasificación, donde datos similares están cercanos entre ellos y lejos de datos diferentes.
- **Fine tunning:** Es un proceso de entrenamiento en el cual como punto de inicio hereda los

pesos de un entrenamiento anterior. Esto ayuda a acelerar el proceso y es especialmente indicado cuando las bases de datos son pequeñas.

También hay que tener en cuenta las mediciones de:

- **Precision at 1 (P@1):** es la media de la precisión del objeto a detectar mejor clasificado obtenido.
- **Precision at 10 (P@10):** Es la media de la precisión de los primeros diez objetos a detectar obtenidos mejor clasificados.
- **Mean average precision (MAP):** Es la media de la precisión promedio, donde la precisión promedio de un análisis de datos única es la media de las puntuaciones de precisión en cada elemento relevante devuelto en una lista de resultados de búsqueda.
- **Mean reciprocal rank (MRR):** Es la media del rango recíproco de la mejor clasificación relevante.
- **R-precision (RP):** Es la media de la precisión después de Rt objetos a detectar para el tema t, donde Rt es el número de objetos a detectar relevantes disponibles para el análisis t

En nuestro caso hemos hecho uso de *MAP@R (Mean Average Precision at R)*, que combina las ideas de *MAP* y *R-precision*.

2.2 Metodología

2.2.1 Tipo de red usada en la segmentación de las bases de datos

La segmentación es el proceso de *deep learning* que asocia una etiqueta o categoría a cada píxel presente en una imagen. En esta investigación se recurre a esto para poder identificar los caminos transitables por coches al clasificar los diferentes elementos de una carretera en estas categorías como: acera, coches, edificios, peatones, árboles, etc. y así poder filtrar la información relevante.

Una de las últimas investigaciones sobre la segmentación para la clasificación de intersecciones es [26] en la que, imágenes obtenidas de Google Maps se usan como base de datos e intentan clasificarlas con cinco clases de intersecciones diferentes. El autor utiliza un sistema de multi-modelo basado en redes neuronales convolucionales para clasificar los tipos de intersecciones. El primer modelo realiza una clasificación binaria de datos entre la parte principal de la imagen y el fondo de la misma. El segundo modelo coge datos obtenidos de la anterior clasificación e intenta aclarar esos casos en las cinco clasificaciones de intersecciones.

Tras esto, crea una red de segmentación semántica tras lo cual, se entrenará la red para clasificar las imágenes en las categorías creadas anteriormente. Si estos procesos funcionan correctamente, la evaluación de esta red será positiva.

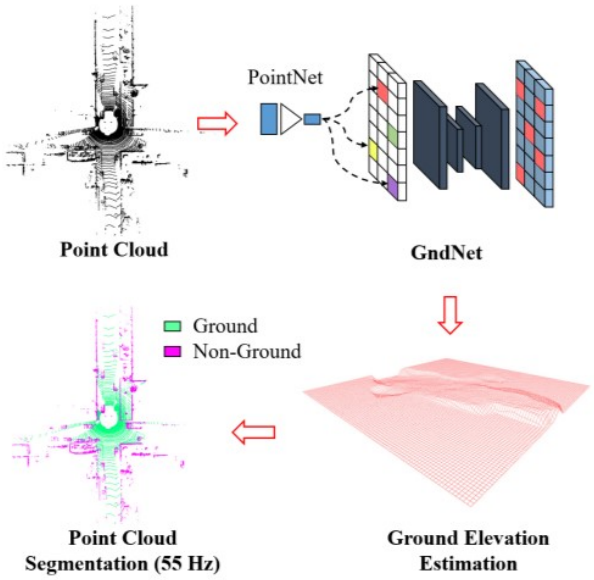


Figura 2.10: Diagrama segmentación GndNet [27]

Otra técnica de segmentación es la llamada *GndNet* usada en [27]. GndNet, acepta puntos sin procesar de nubes de puntos como datos de entrada y produce una estimación de la elevación del terreno y la segmentación de la nube de puntos en dos categorías: suelo y no suelo.

Adicionalmente en la categoría suelo, GndNet diferencia 28 clases de superficies móviles y fijas del terreno, de las cuales solo nos quedaremos con las que indican carretera.

Este proceso de segmentación consiste en en tres pasos como se ilustra en la figura 2.11:

- Primero, se realiza una discretización de la nube de puntos en una red 2D;

- Seguidamente, se agrupan los elementos discretizados anteriormente en pilares (*pillar feature*). Cada pilar contiene todos los elementos de la imagen a las diferentes alturas en el eje Z de cada píxel desde una vista de pájaro.

- Finalmente, una red codificador-decodificador procesa la pseudo-imagen y produce una representación de alto nivel y una regresión de la elevación del suelo por celda, para poder diferenciar lo que es suelo de lo que no lo es (ground/no-ground).

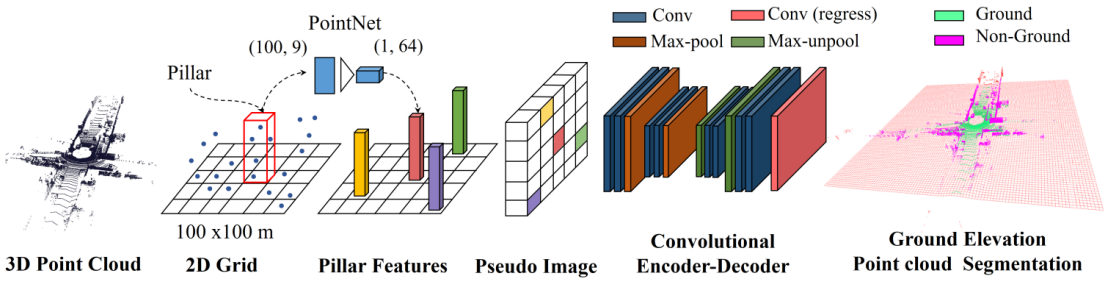


Figura 2.11: arquitectura GndNet [27]

El objetivo de la estimación de *GndNet* es, encontrar la altura del suelo para cada celda de la red, debido a la distribución natural de los puntos de las nubes, este proceso es complicado. La segmentación de los puntos pertenecientes al plano del suelo es también un proceso importante, ya que, si estos puntos se segmentan, tareas como la clasificación o localización de objetos mejorará en velocidad y en precisión.

En este sentido, la investigación con la que se colabora con este TFG, no conseguía una distinción adecuada del suelo. Este TFG persigue así, mejorar la distinción del suelo empleado el método de segmentación recurriendo a las nubes de puntos para diferenciar las superficies de las imágenes. Para ello se emplea el método GndNet.

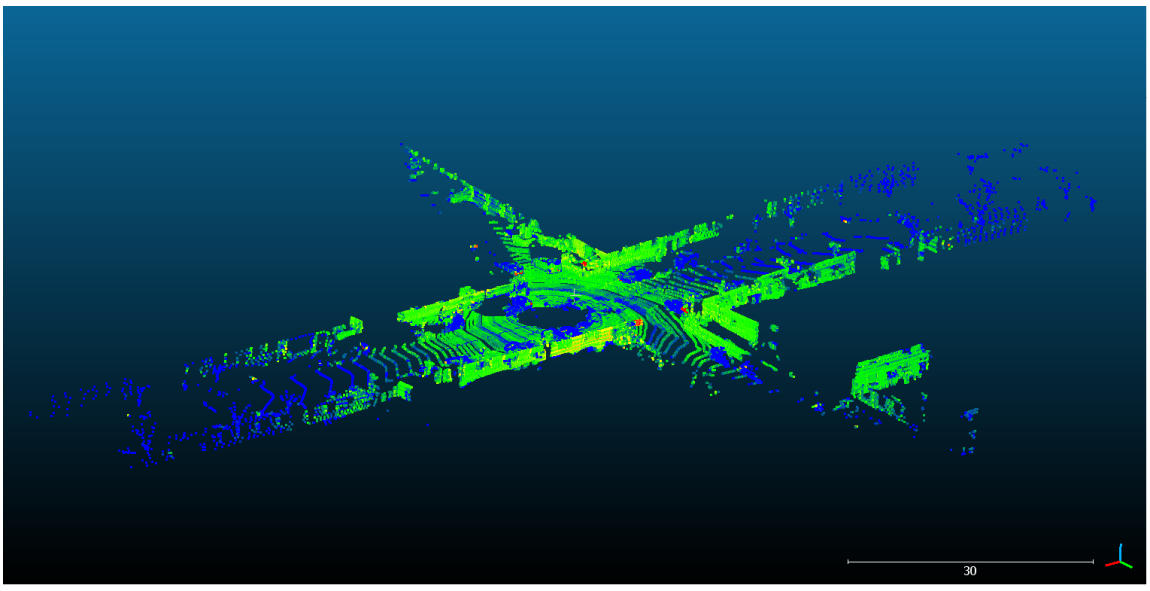


Figura 2.12: Nube de puntos [9]

Como se explicaba anteriormente, *GndNet* acepta nubes de puntos sin procesar como entrada y produce una estimación de la elevación del terreno y una segmentación de la nube de puntos binaria (suelo y no suelo).

La elevación del terreno empleando se puede identificar empleando LIDAR, ya que éste da información de las diferentes superficies que se encuentran a su alcance. En el caso de elementos fijos como paredes o árboles, la elevación de suelo se encuentra donde se encuentra el límite del objeto, como se ilustra en la figura 2.13.

Tras la segmentación que se le aplica a la información de esta base de datos para clasificar las diferentes superficies, se eliminan todos los puntos que no corresponden al suelo y se mantiene solo las de las categorías de carretera, acera, aparcamiento, otros suelos, etc. Después, se calcula la elevación de suelo por celda. Esta tarea sería la de alinear las nubes de puntos y estimar la elevación global del plano para la secuencia completa. Para diferenciar entre suelo y no suelo se emplea el paquete *semantic-KITTI*, base de datos que se explica en 3

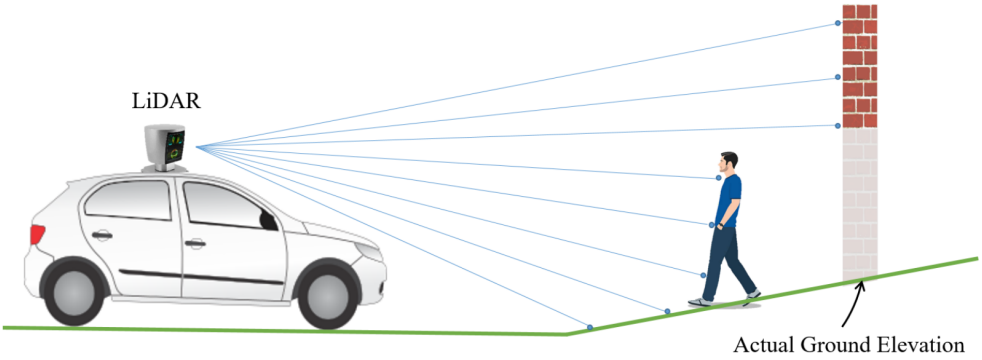


Figura 2.13: elevación de suelo [27]

[9]imagen propia

2.2.2 Entrenamiento de Redes neuronales

En esta investigación se usan redes neuronales y deep learning para entrenar a la máquina, mediante el entrenamiento de las redes. Para el entrenamiento, se han empleado datasets obtenidos de tres fuentes diferentes con diferencias temporales suficientes para evitar en lo posible la correlación entre sí y conseguir así un entrenamiento más eficaz. Una vez entrenado, se procederá a probar con las diferentes bases de datos para comprobar que se consigue crear la diferenciación de la intersección.

Este TFG forma parte de la investigación de análisis y clasificación de próximas intersecciones representada en la figura 2.14. La principal contribución de este trabajo es la de demostrar que, si dejando solo datos de nubes de puntos creadas por LIDAR como datos de entrada en vez de la combinación de estos con imágenes RGB, sigue siendo efectivo el entrenamiento de la red neuronal.

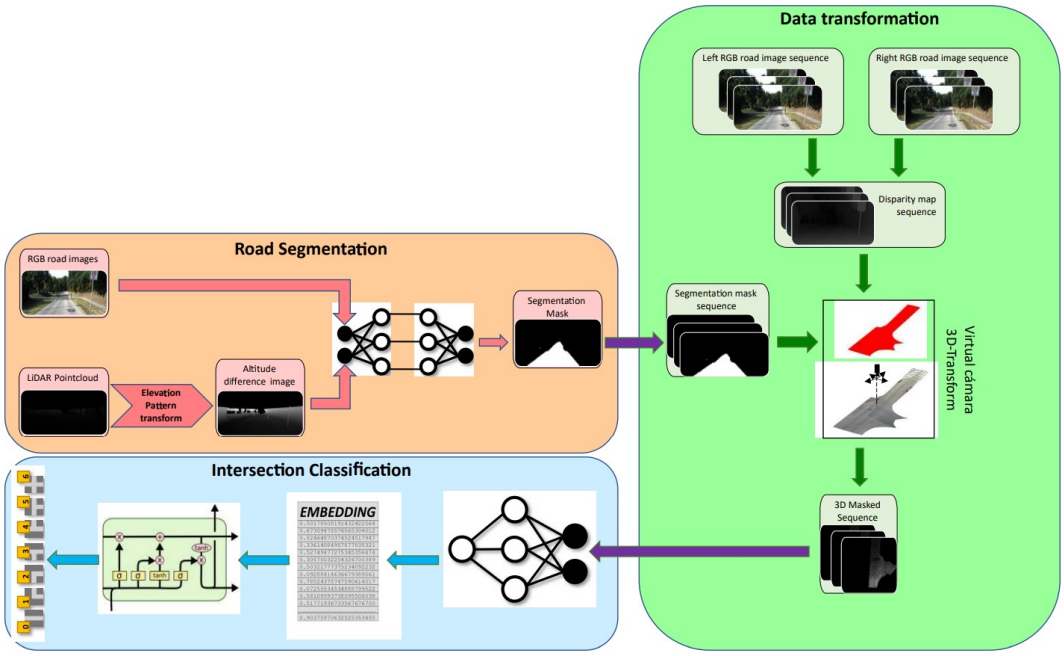


Figura 2.14: Diagrama del grupo de investigación. [28]

La figura 2.14 es el diagrama del proceso desde la segmentación de los datos de la carretera hasta el proceso de entrenamiento de las redes neuronales para conseguir la clasificación de las diferentes intersecciones. Consta de los siguientes bloques:

1. **Segmentación de la carretera:** es el proceso en el que, como se observa en la imagen, mediante dos tipos de bases de datos: imágenes RGB y nubes de puntos obtenidos a través del LIDAR, que se emplean de manera simultanea, se obtiene como salida las diferentes alturas del terreno, es decir, se obtiene la diferenciación de los píxeles que son carretera de los que no lo son.

2. **Transformación de los datos:** En el bloque se procesa la disparidad entre mismas imágenes capturadas con dos cámaras estéreo (dos cámaras separadas para conseguir 3D). Esto está muy relacionado con la estructura 3D, es decir, es como si se calculase la posición 3D de cada píxel de la imagen. Combinando el mapa de disparidad anteriormente mencionado con la máscara

de segmentación creada en el punto anterior, se obtiene la vista de pájaro (vista de planta de la escena a evaluar) que serán los datos de entrada para el entrenamiento.

3. **Clasificación de intersecciones:** En este proceso mediante la utilización de redes neuronales, en concreto Teacher-student explicada en 2.1.2, y con los datos de entrada de la segmentación de la carretera, junto a la información obtenida en la transformación de los datos donde se obtiene la vista de pájaro, se entrenará a la red neuronal para que diferencie entre los siete tipo de intersecciones de clasificación.

Capítulo 3

Descripción del Sistema

3.1 Bases de datos

Se han usado tres bases de datos diferentes, *KITTI y KITTI 365* [29], (Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTI)), estas dos Data-set fueron obtenidas con más de dos años de diferencia, por lo que se puede asegurar así, que la correlación entre estas es baja y por tanto mejoramos la generalización del sistema.

Para evaluar la validez de la clasificación de intersecciones usamos las siguientes bases de datos:

- Base de datos del Grupo *INVETT*: Esta base de datos se creó el 26 de enero de 2021, para la captación de las imágenes se hizo uso de la cámara deportiva 3.1

Figura 3.1: Cámara SJ7-STAR.[1]

- KITTI: Basándose en la investigación de [14], se seleccionan ocho secuencias de zonas residenciales, todas ellas grabadas en el 2011. Las imágenes fueron automáticamente escogidas de toda la secuencia seleccionando solo aquellas que estuviesen a menos de 20m de una intersección. El principal problema es, que esta base de datos tiene relativamente pocas imágenes y el reparto entre las diferentes clases de intersecciones no es equilibrado.
- KITTI 360: Esta base de datos contiene diez nuevas secuencias grabadas en 2013, casi dos años después de que se grabasen las de la base de datos anterior. Las imágenes están etiquetadas

[1] imagen extraída de: https://www.sjcam.com/es/producto/sj7-waterproof-motorcycle-case/

incluyendo solo aquellas que contienen intersecciones usando las imágenes de la base de datos anterior como referencia.

Figura 3.2: Imagen de muestra de los Datasets de KITTI [2]

Para la realización de las bases de datos de KITTI, se utilizó como se mustra en la figura 3.4

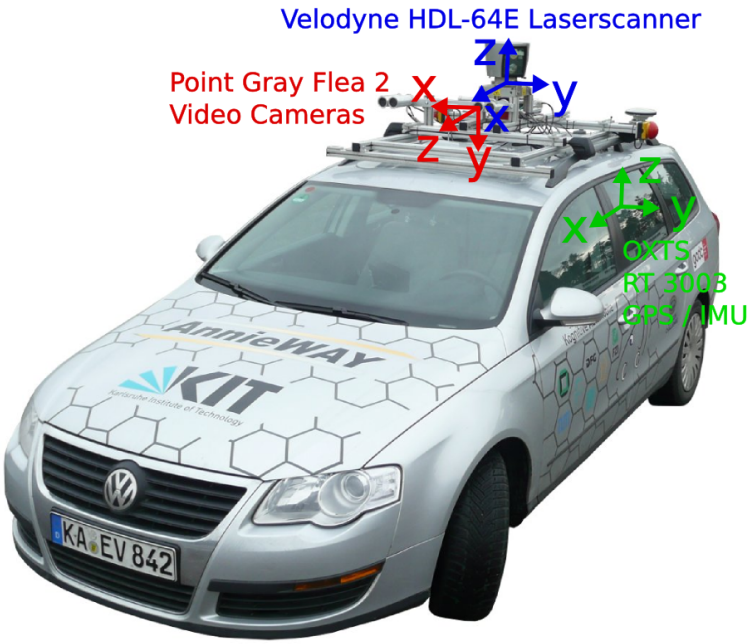


Figura 3.3: VW Passat sensors [3][30]

El vehículo está equipado de los siguientes dispositivos:

- 1 Sistema de navegación (GPS/IMU): OXTS RT 3003
- 1 LIDAR: Velodyne HDL-64E
- 2 Cámaras ByN, 1.4 Megapixels: Point Grey Flea 2 (FL2-14S3M-C)
- 2 Cámaras en color, 1.4 Megapixels: Point Grey Flea 2 (FL2-14S3C-C)
- 4 Objetivos varifocales, 4-8 mm: Edmund Optics NT59-917

[2]imagen extraída de base de datos KITTI
[3]http://www.cvlibs.net/datasets/kitti/
[4]http://www.cvlibs.net/datasets/kitti/

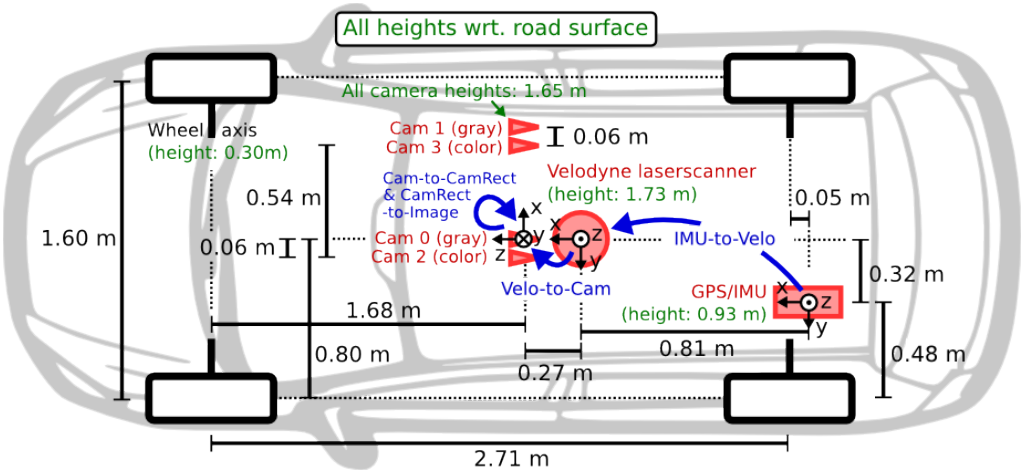


Figura 3.4: Vistas del vehículo KITTI Dataset.[4][30]

3.2 Equipo para el entrenamiento y creación de código

Dado que en este TFG se colabora con *INVETT* se ha hecho uso del PC instalado en el laboratorio del grupo de investigación, y mediante la VPN de la Universidad y el acceso al mismo desde mi portátil, se programó en remoto.

Estos equipos constan de las siguientes características:

Equipo de la Universidad:

- Microprocesador : I3 2100 @3.100GHz
- GPU: Nvidia 1080 Ti 12GB

Equipo personal:

- Microprocesador: I5 6200 @2.400GHz
- RAM: 8GB

Capítulo 4

Pruebas

4.1 Procesado de imágenes

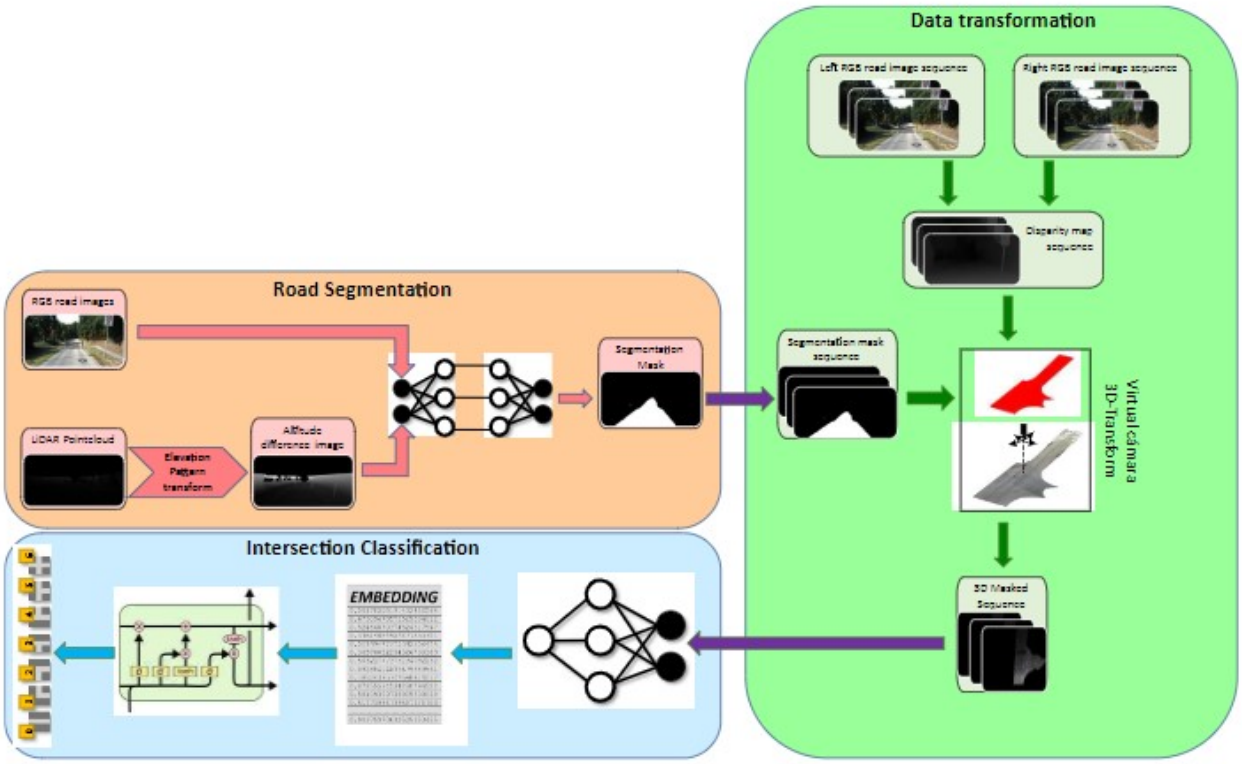


Figura 4.1: workflow de investigación [28]

Mediante las pruebas que se explican en este capítulo queremos ver la eficacia de entrenar la red neuronal con una base de datos creada con nubes de puntos, la cual ha sido creada previamente mediante datos del *LIDAR* en lugar de mediante las imágenes obtenidas por cámaras.

Como se explica en 2.2, los datos recibidos del *LIDAR* son procesados mediante un proceso de segmentación en el que se filtran los puntos repetidos y se discriminan según la localización en el eje Z, así se consigue diferenciar lo que es carretera de lo que no lo es.

Esto resulta en una representación en 3D de la nube de puntos. Ya que se quiere simplificar las variables que entren como información, a las redes neuronales y hacer más sencilla la base de datos, se procesará la imagen para conseguir una vista de pájaro (vista de planta XY) de la nube de puntos. Posteriormente, se orientará con la vista frontal del vehículo representada en la parte positiva del eje Y, para crear así imágenes.

Puesto que el objetivo es el de comparar con las imágenes obtenidas de las cámaras estéreo instaladas en el techo del vehículo y ver así, si se consigue obtener mejores resultados.

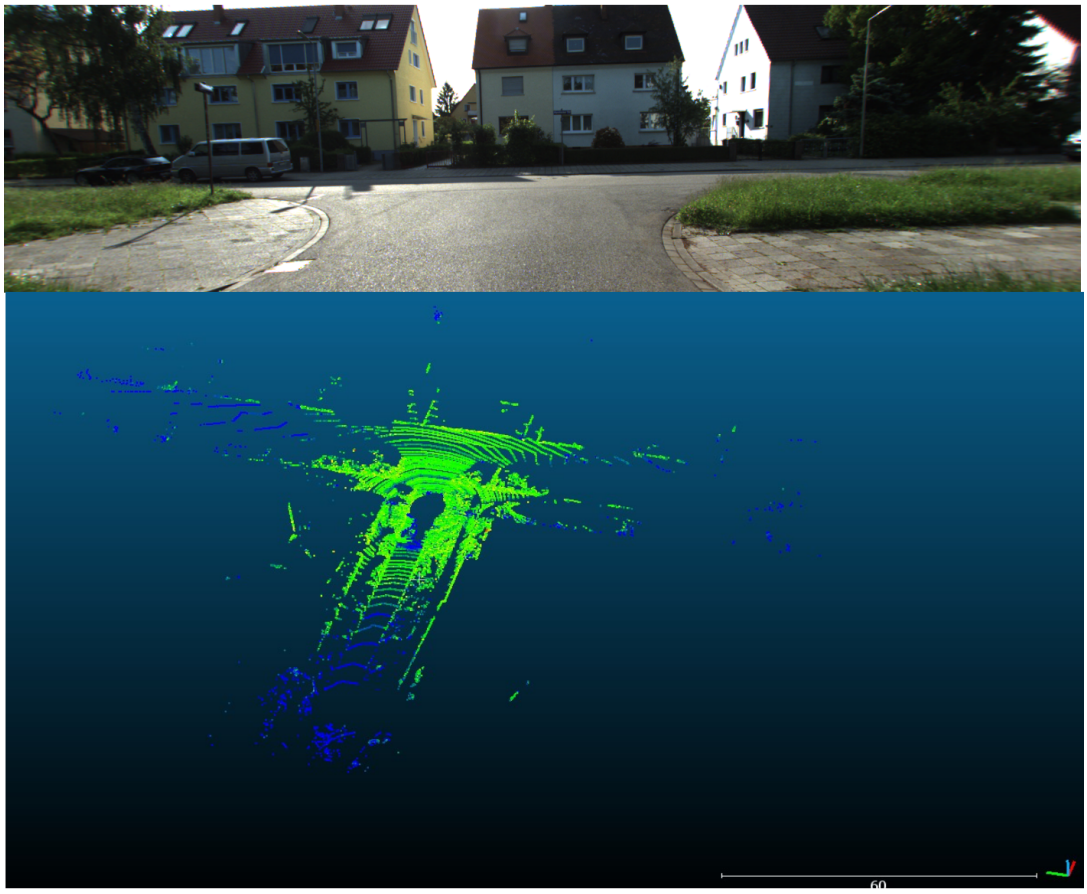


Figura 4.2: Comparación de imagen con nube de puntos [1]

La imagen 4.2 es la comparación de la fotografía y la imagen sintética generada tras el procesamiento completo (generación de la nube de puntos con LIDAR, segmentación mediante GndNet y filtrado para la obtención del suelo).

[1] composición propia superponiendo la imagen extraída de KITTI junto a la nube de puntos final tras la segmentación

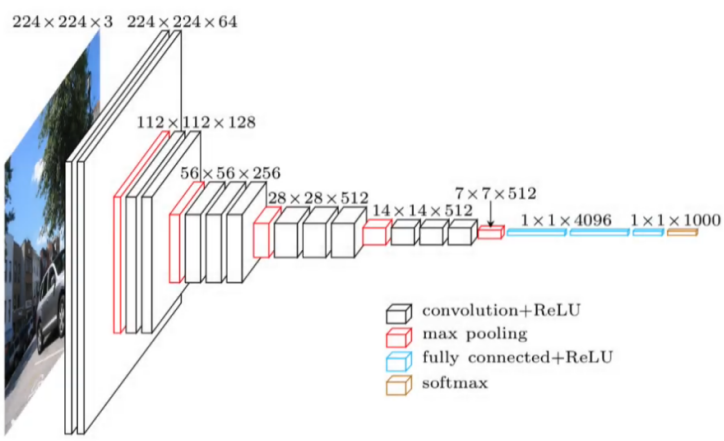


Figura 4.3: dimensiones de datos de entrada [2][31]

Tras esto, se ha de adecuar la resolución de la imagen a los parámetros de entrada de la red neuronal. Para ello, como se observa en la imagen 4.3 la se elegirá la resolución de las imágenes creadas, en nuestro caso de 224x224, ya que es el tamaño que tienen el resto de imágenes y la red neuronal está configurada para recibir bases de datos con esta resolución de entrada (se podría, si se quisiese, configurar a otro tamaño de imagen).

Finalmente, se hace una limitación de las distancias hasta donde queremos que los puntos sean relevantes. Basándonos en investigaciones pasadas [32] y tras valorar diferentes distancias, finalmente se decide elegir 20m a los lados y entre 2 y 20m hacia el frontal del vehículo ya que la parte de atrás no es relevante para este estudio y mas allá de 20m los datos son innecesarios pues que se evalúan los cruces cercanos.

En las imágenes 4.4, la imagen izquierda muestra el resultado tras los límites propuestos y la derecha la imagen XY sin los límites.

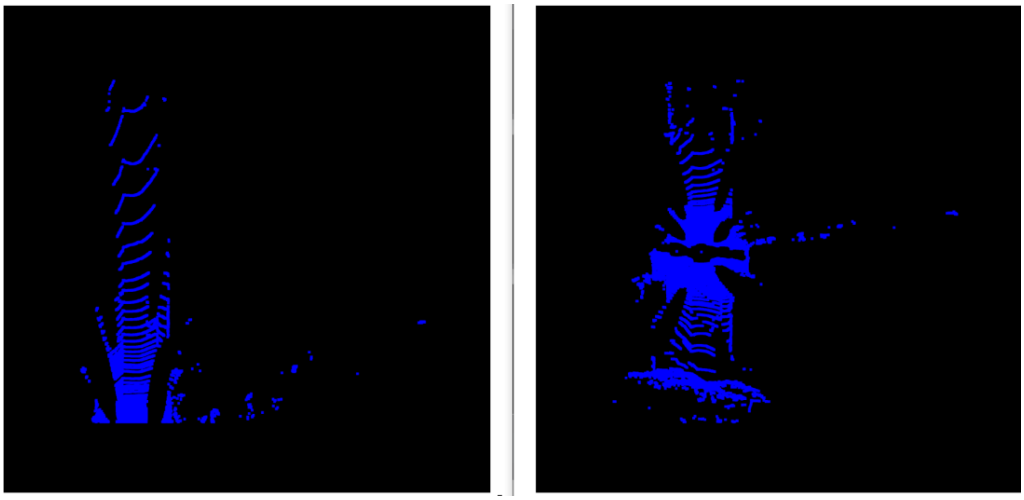


Figura 4.4: Ilustración de los límites propuestos [3]

[2]https://www.techleer.com/articles/305-vgg-16-an-advanced-approach-towards-accurate-large-scale-image-recognition/

[3]imagen propia

Capítulo 5

Resultados

Uno de los objetivos de este TFG era averiguar si el uso de las imágenes creadas a partir de los datos obtenidos del LIDAR proporcionan resultados adecuados a la hora de entrenar a la red neuronal.

Para esto se creó una base de datos de imágenes a partir de los datos proporcionados por el LIDAR, y para re-entrenar la red neuronal *Teacher-student*, se empleó el proceso *Fine tuning*, variando el *LR, el optimizador y la arquitectura de red* entre otros hiper-parámetros para intentar obtener unos resultados óptimos.

Con los resultados obtenidos de los diferentes entrenamientos se generó la tabla 5.1 con los valores correspondientes de la variable MAP@R, que es la variable más indicada para representar la diferenciación entre las diversas clases de intersecciones.

La tabla 5.1 recoge los resultados de los entrenamientos con el diferentes valores asignados para el *learning rate*, para intentar mejorar la variable de MAP@R.

En estas tablas se puede observar que, ajustando el *Learning Rate*, se puede mejorar los valores obtenidos para MAPR. Sin embargo, a pesar de ello, con ningún valor se alcanzan valores relevantes MAP@R para poder afirmar que la red neuronal esta aprendiendo correctamente con esta base de datos, ya que en ninguno de los casos se obtienen valores lo suficientemente altos.

Learning Rate	Optimizer	Val/MAPR
0,00065	Adamax	0,32
0,05	Adam	0,15
0,04	Adam	0,14
0,003	Adam	0,19
0,005	Adam	0,18
0,004	Adam	0,27
0,0048	Adam	0,287
0,0049	Adam	0,17
0,0052	Adam	0,18
0,006	Adam	0,18
0,002	Adam	0,23
0,001	Adam	0,265
0,0009	Adam	0,27
0,0008	Adam	0,27
0,0007	Adam	0,29
0,0006	Adam	0,3
0,0005	Adam	0,315
0,0004	Adam	0,31
0,0003	Adam	0,32
0,0002	Adam	0,32
0,0001	Adam	0,315
0,00009	Adam	0,315
0,00008	Adam	0,33
0,00007	Adam	0,31
0,00006	Adam	0,33
0,00005	Adam	0,32
0,00004	Adam	0,31
0,00003	Adam	0,32
0,00002	Adam	0,31
0,00001	Adam	0,28
0,000009	Adam	0,27
0,000008	Adam	0,265
0,000007	Adam	0,31
0,000006	Adam	0,22
0,0000006	Adam	0,21

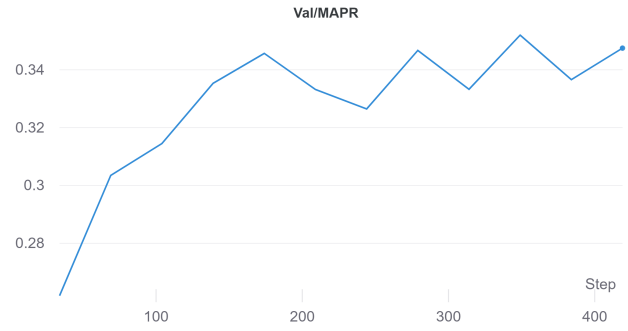
Tabla 5.1: Learning Rate Val/MAPR

En la tabla 5.2 se recogen los mejores valores de Val/MAP@R que se obtuvieron con un LR de 0,00006, empleando los diferentes optimizadores. Sin embargo, se comprueba que en ningún caso se llega a alcanzar un valor al menos de 0,4.

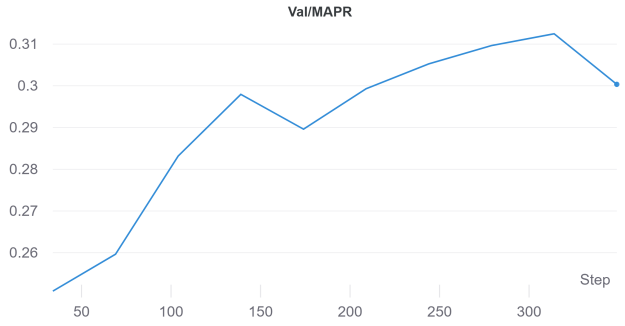
Learning Rate	Optimizer	Val/MAPR
0,00006	AdamMax	0,31
0,00006	adamW	0,3
0,00006	ASDGD	0,29
0,00006	SGD	0,27
0,00006	rmsprop	0,25

Tabla 5.2: Same Learning Rate using Different Optimizers and Val/MAPR

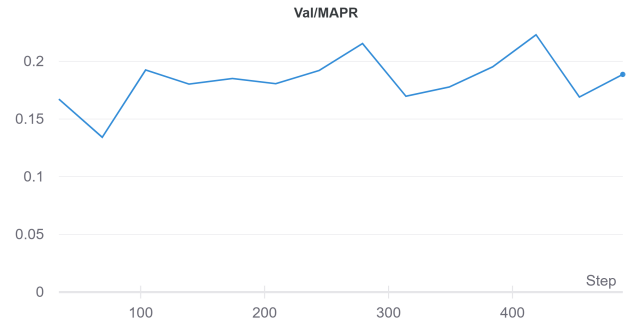
Las figuras 5.1 y 5.2 indican los valores de Val/MAPR obtenidos de diferentes entrenamientos. Ningún resultado supera los 0,4 lo cual indica que los entrenamientos no están siendo efectivos ya que indica que no está consiguiendo clasificar efectivamente lo diferentes tipos de intersecciones.



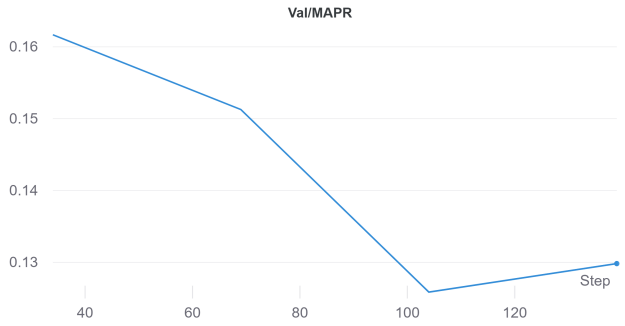
(a) 0.34



(b) 0,3004



(c) 0.1887



(d) 0,1298

Figura 5.1: MAPR

Existen múltiples razones por las que un entrenamiento no entrega los resultados deseados [33], [34]. Entre otras, las más importantes son:

- El modelo puede estar "subestimado". es decir, el proceso de optimización se detuvo de forma temprana. En este caso podría ser posible aún, obtener un menor error de ajuste a la muestra de calibración continuando el entrenamiento. Esto se puede observar en las curvas de pérdidas, en las que, o bien se observa que la gráfica tiene una zona plana o mucho ruido, indicando que

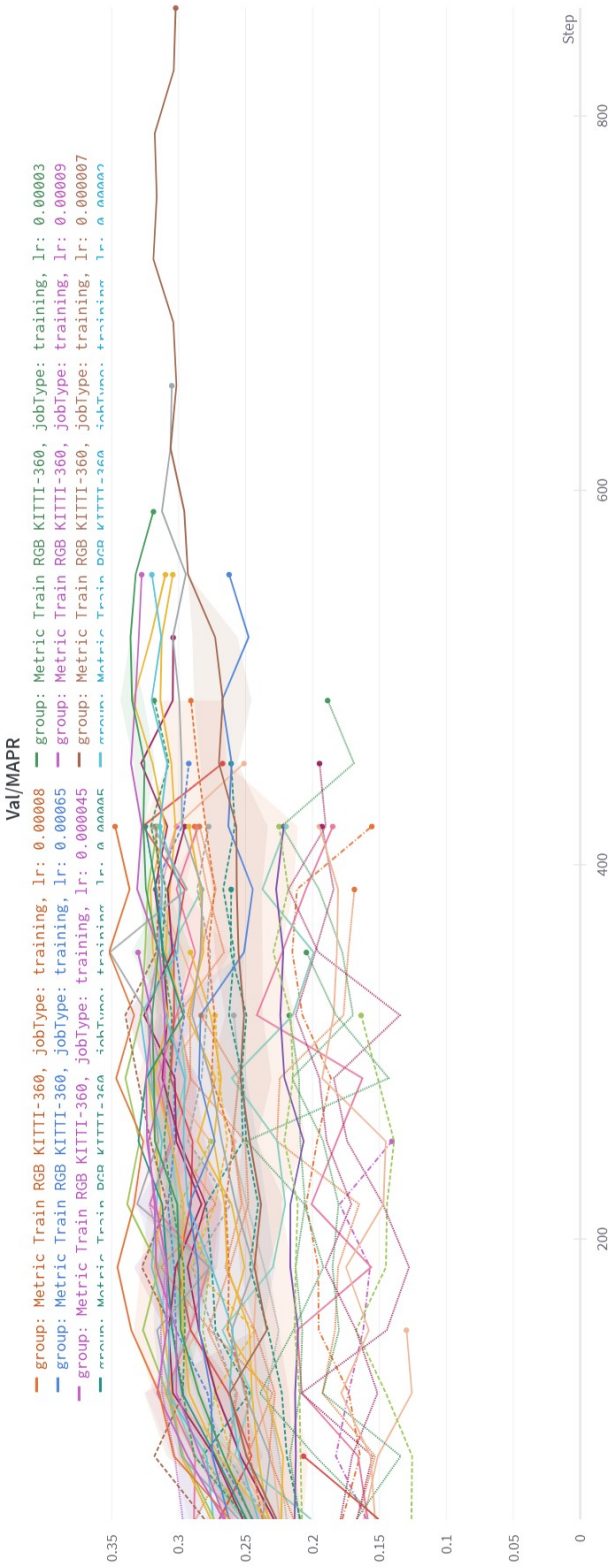


Figura 5.2: Comparación de batch training Val/MAPR de 70 entrenamientos

está mal dimensionado el modelo para la complejidad del sistema; o bien, la curva al final de la gráfica sigue con una tendencia descendente, indicando que necesita más epochs para seguir aprendiendo o que el LR está demasiado bajo.

- Sobre-entrenamiento. El modelo ha memorizado los datos de calibración (entrenamiento), lo que indica que ha dejado de generalizar. De tal manera al mostrar una precisión muy alta, podría parece que el entrenamiento ha tenido éxito. Sin embargo, la gráfica de pérdidas tiene forma de U, evidencia que no se produce aprendizaje sino memorización.

  Esto se puede solucionar disminuyendo el LR para reducir la velocidad de aprendizaje; regulando la capacidad del modelo reduciendo del número o tamaño de sus capas ocultas; agregando un early stopping para parar el entrenamiento cuando la curva de validación muestra que se ha dejado de mejorar; o regularizando los pesos para limitar la complejidad de la red. Normalmente el sobre-entrenamiento se puede reducir pero nunca eliminar.

  El entrenamiento óptimo se consigue cuando el error de ajuste disminuye progresivamente a medida que se aumenta la complejidad de la red neuronal, a tal punto que sea arbitrariamente cercano a cero si la complejidad del modelo es la suficiente.

  Tras estas explicaciones, en la figura 5.3 se pueden ver representados 70 entrenamientos. En todos los casos se observa una cantidad de ruido muy alto. Esto podría ser debido a cuestiones como estas:

  – Un *LR* muy bajo. Sin embargo, se observa que el rango de análisis del *LR* es desde 0,55 hasta 0,0000005, que es suficientemente amplio.
  – Poco número de repeticiones. Sin embargo, no es el caso ya que se ha probado tanto con *early stopping* como sin él, aumentando considerablemente el número de *epochs*, evidenciando que no influye en el resultado al no seguir una curva de entrenamiento correcta.

  Una primera interpretación podría ser que, el problema del entrenamiento mencionado se deba a la falta de paquetes de datos, ya que el número de bases de datos empleadas es insuficiente.

  De la figura 5.4 se observa que de los 70 entrenamientos, hay muchos que parece que aprenden demasiado bien, esto es por una clara memorización en vez de una generalización que es, lo que debería hacer un entrenamiento correcto.

  En las gráficas 5.3 y 5.4, se observa una muy alta dispersión entre los valores de cada entrenamiento, reflejando así un alto ruido.

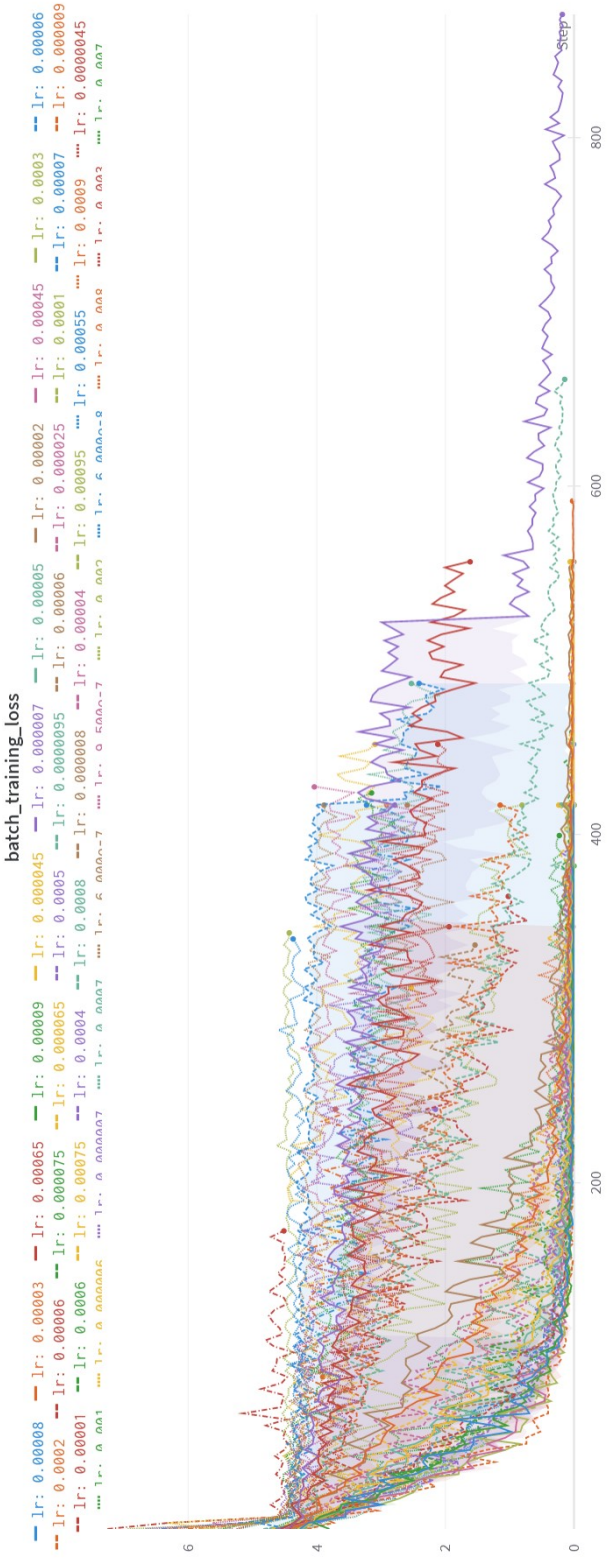


Figura 5.3: Comparación de batch training loss de 70 entrenamientos

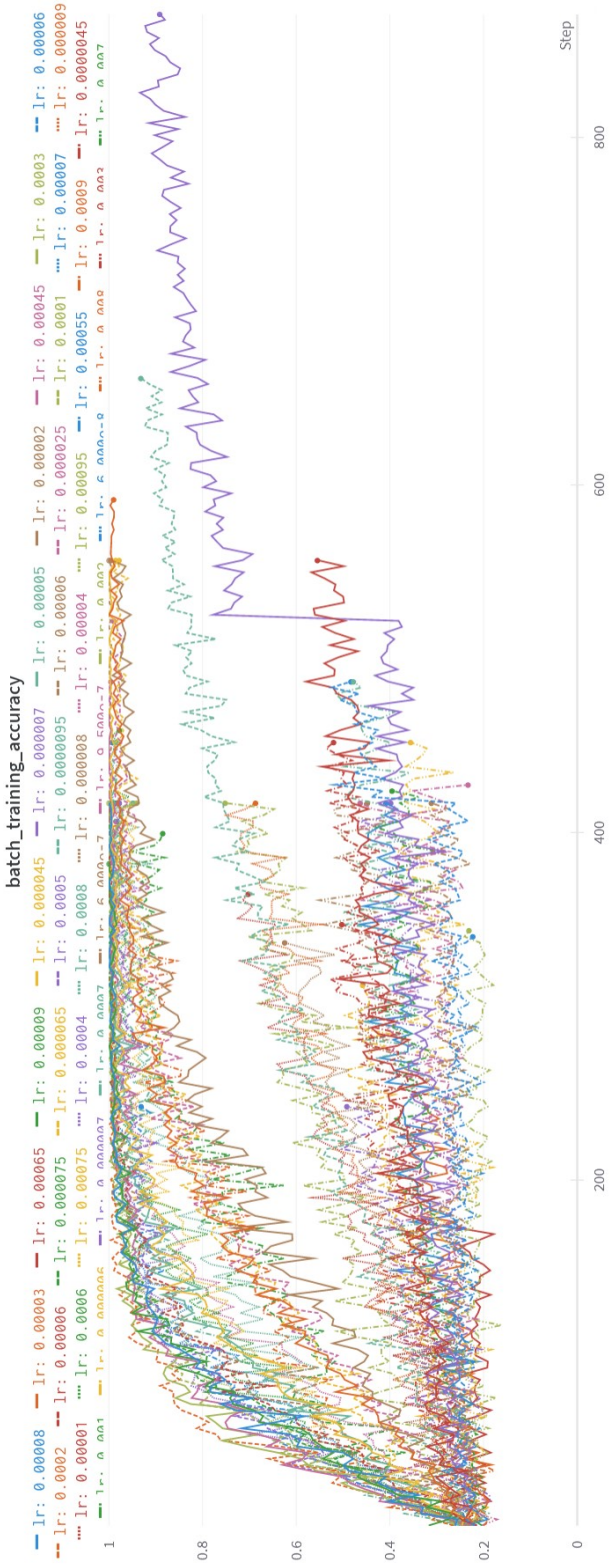


Figura 5.4: Comparación de batch training accurancy de 70 entrenamientos

Capítulo 6

Conclusiones y líneas futuras

Los resultados obtenidos con este estudio muestran que el entrenamiento no genera un correcto aprendizaje de la red neuronal, evidenciándose este hecho en los bajos valores de MAP@R y a la alta dispersión (ruido) en los resultados de los entrenamientos.

Estos resultados pueden deberse a cuestiones como que el número de bases de datos es insuficiente y adicionalmente al contener éstas imágenes de carreteras europeas, que son particularmente estrechas, contienen un alto nivel de ruido.

Como lección aprendida para futuras líneas de investigación, podríamos proponer que gracias al creciente número de investigaciones en este campo, se puede contar cada vez con un mayor número y diversidad de bases de datos. Esto posibilitará escoger mejor su número y tipo, evitando así que los datos estén correlacionados entre sí y facilitando así un aprendizaje más efectivo.

6.1 Comentarios finales

Gracias a esta investigación, he tenido la oportunidad de adquirir nuevos y variados conocimientos que abarcan desde el empleo de nuevas herramientas de edición (LaTEX), lenguajes de programación empleados en el campo de las redes neuronales (phyton-phytorch), hasta la adquisición de nuevos conocimientos en el campo de la inteligencia artificial como es el deep learning, visión artificial (imágenes LIDAR), redes neuronales en general y en especial las convolucionales, sus tipologías, configuración y el manejo de sus híper-parámetros, y en particular en su aplicación a problemas de detección de objetos en imágenes.

Bibliografía

[1] A. L. Ballardini, D. Cattaneo, y D. G. Sorrenti, "Visual localization at intersections with digital maps," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6651–6657.

[2] D.-G. for Mobility y Transport, "fatalities between pricipal users," *European Commission*, vol. 58, no. 12, noviembre 2021. [Online]. Disponible en: https://transport.ec.europa.eu/news/road-safety-european-commission-rewards-effective-initiatives-and-publishes-2020-figures-road-2021-11-18_en

[3] M. Gursoy y E. Korkmaz, "Statistical analysis of traffic accidents in the küçükçekmece district of İstanbul," vol. 38, pp. 1869–1878, 12 2020.

[4] A. GUTIÉRREZ, "Las intersecciones en forma de 'x', las más peligrosas," *Appl. Opt.*, vol. 58, no. 12, marzo 2018. [Online]. Disponible en: https://revista.dgt.es/es/noticias/nacional/2018/03MARZO/0321un-40-por-ciento-de-los-accidentes-con-victimas-ocurren-en-intersecciones.shtml

[5] Maria Eugenia Rivas Amiassorho, "`https://blogs.iadb.org/transporte/wp-content/uploads/` 27 de Junio de 2020.

[6] BRISKA, "`https://biriska.com/conoces-los-modernos-sistemas-adas-de-los-coches/`," 21 de Noviembre de 2018.

[7] V. Fuentes, "Así fue como en 1917 la doctora june mccarroll introdujo las marcas viales para mejorar la seguridad de las carreteras," *motorpasión*, vol. 1, p. 1, 05 2020.

[8] Jaime Ramos, "`https://www.circulaseguro.com/inc/uploads/2020/07/`," 31 de Julio de 2020.

[9] GONZALO GARCÍA, "`https://www.hibridosyelectricos.com/media/hibridos/`," 11 de Marzo de 2021.

[10] J. An, B. Choi, K.-B. Sim, y E. Kim, "Novel intersection type recognition for autonomous vehicles using a multi-layer laser scanner," *Sensors*, vol. 16, no. 7, 2016. [Online]. Disponible en: https://www.mdpi.com/1424-8220/16/7/1123

[11] D. Bhatt, D. Sodhi, A. Pal, V. Balasubramanian, y M. Krishna, "Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras," in *2017*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 4495–4500.

[12] *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*. IEEE, 2019. [Online]. Disponible en: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8792328

[13] U. Baumann, Y. Huang, C. Gläser, M. Herman, H. Banzhaf, y J. M. Zöllner, "Classifying road intersections using transfer-learning on a deep neural network," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, November 2018, pp. 683–690.

[14] A. L. Ballardini, D. Cattaneo, S. Fontana, y D. G. Sorrenti, "An online probabilistic road intersection detector," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 239–246.

[15] systems, "`https://magiquo.com/redes-neuronales-o-el-arte-de-imitar-el-cerebro-huma`
1 de Noviembre de 2019.

[16] E. Shelhamer, J. Long, y T. Darrell, "Fully convolutional networks for semantic segmentation," 2016. [Online]. Disponible en: https://arxiv.org/abs/1605.06211

[17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, y Y. Wei, "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 764–773.

[18] G. E. Hinton, O. Vinyals, y J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015. [Online]. Disponible en: http://arxiv.org/abs/1503.02531

[19] D. Yu, K. Yao, H. Su, G. Li, y F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. IEEE, 2013, pp. 7893–7897. [Online]. Disponible en: https://doi.org/10.1109/ICASSP.2013.6639201

[20] J. Li, M. L. Seltzer, X. Wang, R. Zhao, y Y. Gong, "Large-scale domain adaptation via teacher-student learning," *CoRR*, vol. abs/1708.05466, 2017. [Online]. Disponible en: http://arxiv.org/abs/1708.05466

[21] systems, "`https://github.com/ShivamRajSharma/Teacher-Student-Network//`," 6 de Julio de 2021.

[22] D. Wang y T. F. Zheng, "Transfer learning for speech and language processing," *CoRR*, vol. abs/1511.06066, 2015. [Online]. Disponible en: http://arxiv.org/abs/1511.06066

[23] A. Golatkar, A. Achille, y S. Soatto, "Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence," 2019.

[24] H. Apaydin, H. Feizi, M. Sattari, M. S. Çolak, S. Band, y K.-W. Chau, "Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting," *Water*, 05 2020.

[25] systems, "https://towardsdatascience.com/review-inception-v4-evolved-from-googlen
27 de Septiembre de 2018.

[26] C.-L. Kuo y M.-H. Tsai, "Road characteristics detection based on joint convolutional neural networks with adaptive squares," *ISPRS International Journal of Geo-Information*, vol. 10, no. 6, 2021. [Online]. Disponible en: https://www.mdpi.com/2220-9964/10/6/377

[27] A. Paigwar, Ö. Erkent, D. S. González, y C. Laugier, "Gndnet: Fast ground plane estimation and point cloud segmentation for autonomous vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[28] ""a phd dissertation on road topology classification for autonomous driving," Ph.D. dissertation, 9 2021.

[29] A. Geiger, P. Lenz, C. Stiller, y R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[30] A. Geiger, P. Lenz, y R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[31] Prakarsh Saxena, "https://www.techleer.com/articles/305-vgg-16-an-advanced-approach
recognition//," 6 de Septiembre de 2017.

[32] A. L. Ballardini, D. Cattaneo, S. Fontana, y D. G. Sorrenti, "An online probabilistic road intersection detector," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 239–246.

[33] J. Brownlee, "How to use learning curves to diagnose machine learning model performance," vol. 12, pp.–, 2 2019.

[34] ——, "Diagnosing model performance with learning curves," vol. 3, pp.–, 2 2013.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá