UNIVERSIDAD DE SEVILLA

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS

Técnicas de Inteligencia Artificial Aplicadas a Sistemas de Detección y Clasificación de Señales de Tráfico

Tesis Doctoral presentada por

D. Álvaro Arcos García

dirigida por

Dr. Juan Antonio Álvarez García

y **Dr. Luis Miguel Soria Morillo**.

Septiembre 2018

A mi familia.

Por hacer siempre todo lo posible para hacerme feliz.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a mis directores de Tesis, Dr. D. Juan Antonio Álvarez García y Dr. D. Luis Miguel Soria Morillo, por todo el apoyo, tiempo y esfuerzo que le han dedicado a la elaboración de este trabajo. Me habéis ayudado a crecer tanto personal como profesionalmente durante estos años, gracias por confiar en mí. En especial a Juan Antonio, te estaré eternamente agradecido por estar siempre pendiente de mí, preocupándote por todo lo que pasaba a mi alrededor, y ofreciéndome siempre tu apoyo incondicional.

A toda mi familia y amigos, tanto los que están como los que se fueron, por haberme apoyado siempre en todas las decisiones de mi vida sin poner nunca ninguna objeción, y haberme inculcado valores como el trabajo, el esfuerzo y la superación, los cuales me han permitido alcanzar grandes objetivos en mi vida. Os quiero.

A los compañeros del Departamento de Lenguajes y Sistemas Informáticos, en especial a José María Luna y José Antonio Fábregas. No me canso de decirlo, sois dos grandes amigos y habéis sido un pilar fundamental en los momentos difíciles, preocupándoos por mí, sacándome sonrisas y aconsejándome siempre lo mejor. Todos esos momentos, y a vosotros como persona, jamás lo olvidaré. También me gustaría agradecer a Pepe Riquelme Santos y a Jorge García Gutiérrez haberme dado la oportunidad de trabajar en vuestro grupo de investigación, así como por los grandes consejos que me habéis dado durante esta etapa.

A todos, muchas gracias de corazón.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Resumen

Esta tesis, presentada como conjunto de artículos de investigación, estudia y analiza soluciones para los sistemas de detección y clasificación de señales de tráfico que suponen un reto en aplicaciones de la actualidad, como son la seguridad y asistencia en carretera a conductores, los coches autónomos, el mantenimiento de señalización vertical, o el análisis de escenas de tráfico.

Las señales de tráfico constituyen un activo fundamental dentro de la red de carreteras porque su objetivo es ser fácilmente perceptible por los peatones y conductores para advertirles y guiarlos tanto de día como de noche. El hecho de que las señales estén diseñadas para ser únicas y tener características distinguibles, como formas simples y colores uniformes, implica que su detección y reconocimiento sea un problema limitado. Sin embargo, el desarrollo de un sistema de reconocimiento de señales en tiempo real aún presenta desafíos debido a los tiempos de respuesta, los cuales son cruciales para tomar decisiones en el entorno, y la variabilidad que presentan las imágenes de escenas de tráfico, que pueden incluir imágenes a distintas escalas, puntos de vista complicados, oclusiones, y diferentes condiciones de luz. Cualquier sistema de detección y clasificación de señales de tráfico debe hacer frente a estos retos.

En este trabajo, se presenta un sistema de clasificación de señales de tráfico basado en aprendizaje profundo (Deep Learning). Concretamente, los principales componentes de la red neuronal profunda (Deep Neural Network) propuesta, son capas convolucionales y redes de transformaciones espaciales (Spatial Transformer Networks). Dicha red es alimentada con imágenes RGB de señales de tráfico de distintos países como Alemania, Bélgica o España. En el caso de las señales de Alemania, que pertenecen al dataset denominado German Traffic Sign Recognition Benchmark (GTSRB), la arquitectura de red y los parámetros de optimización propuestos obtienen un 99.71 % de precisión, mejorando tanto al sistema visual humano como a todos los resultados previos del estado del arte, siendo además más eficiente en términos de requisitos de memoria. En el momento de redactar esta tesis, nuestro

método se encuentra en la primera posición de la clasificación a nivel mundial.

Por otro lado, respecto a la problemática de la detección de señales de tráfico, se analizan varios sistemas de detección de objetos propuestos en el estado del arte, que son específicamente modificados y adaptados al dominio del problema que nos ocupa para aplicar la transferencia de conocimiento en redes neuronales (transfer learning). También se estudian múltiples parámetros de rendimiento para cada uno de los modelos de detección con el fin de ofrecer al lector cuál sería el mejor detector de señales teniendo en cuenta restricciones del entorno donde se desplegará la solución, como la precisión, el consumo de memoria o la velocidad de ejecución. Nuestro estudio muestra que el modelo Faster R-CNN Inception Resnet V2 obtiene la mejor precisión (95.77 % mAP), mientras que R-FCN Resnet 101 alcanza el mejor equilibrio entre tiempo de ejecución (85.45 ms por imagen) y precisión (95.15 % mAP).

PARTE I

Prefacio

CAPÍTULO 1

INTRODUCCIÓN

Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time. - Thomas Alva Edison

1.1. Motivación de la investigación

De acuerdo con la Federación Europea de Carreteras (ERF), existe una tendencia negativa con respecto a las inversiones en infraestructura vial y mantenimiento, ya que la financiación de esos gastos está disminuyendo desde 2008 [12]. Este reporte señala que esta tendencia tiene un impacto económico masivo a medio y largo plazo, ya que tanto las inversiones requeridas para el mantenimiento de la infraestructura como el coste de los vehículos necesarios para realizar dicha tarea, aumentan exponencialmente a medida que la condición de la carretera se deteriora.

Las señales de tráfico son un activo esencial que regula el tráfico y guía tanto a conductores como a peatones. Debido a esto, las inspecciones periódicas deben garantizar que la visualización de las señales sea correcta. Sin embargo, el ERF señaló la existencia de un retraso alarmante en el mantenimiento de señales de tráfico en muchos países europeos que reduce la seguridad de las carreteras ya que las señales de tráfico podrían tener colores deteriorados o perder sus propiedades reflectivas. Dado que los accidentes causados por deficiencias en la infraestructura de las carreteras resultan en altos costes humanos y económicos, invertir en dicha infraestructura (especialmente en señalización vertical) tiene un impacto positivo en términos de seguridad vial y rentabilidad económica. Existen diferentes estrategias para el mantenimiento y la sustitución de las señales de tráfico. Se pueden reemplazar en intervalos de tiempo fijos o periódicos. Por lo general, se llevan a cabo de forma in situ y manual.

Hoy en día, la tecnología basada en sensores remotos permite que las carreteras sean analizadas con mayor rapidez, seguridad y con un menor gasto de recursos, lo que mejora significativamente los resultados de las inversiones en infraestructuras viales. Los Sistemas de Mapeo Móvil (MMS) son capaces de recolectar grandes cantidades de datos 3D y 2D utilizando la tecnología Mobile Laser Scanner (MLS) junto con sistemas de imágenes. Las representaciones 3D de los entornos escaneados son densas, precisas, y proporcionan información relevante. Sin embargo, a pesar de la creciente atención que está recibiendo esta tecnología, existen limitaciones dadas por la resolución del sistema de escaneo y la necesidad de altas capacidades de almacenamiento y procesamiento.

En el sector de la conducción autónoma, la detección y la clasificación de señales de tráfico es un pilar fundamental para conseguir un nivel de independencia real del conductor, y por lo tanto, es un tema de investigación actual basado en los campos de la visión por computador y la inteligencia artificial. El desarrollo de un sistema robusto de reconocimiento de señales de tráfico que funcione en tiempo real es aún una tarea desafiante debido a la variabilidad del mundo real, como por ejemplo variaciones de escala o tamaño de las señales en las imágenes, puntos de vista

complejos, desenfoques debido al movimiento, decoloración de las señales, oclusiones y distintas condiciones de luz. Además, hay más de 300 categorías diferentes de señales de tráfico definidas por la Convención de Viena sobre el tráfico y carreteras [78]. Este tratado ha sido firmado por 63 países, aunque existen algunas variaciones visuales menores de pictografías de señales de tráfico, lo cual supone una mayor complicación para la tarea de reconocimiento automatizado.

Esta tesis doctoral presenta el trabajo desarrollado en técnicas de inteligencia artificial aplicadas a sistemas de detección y clasificación de señales de tráfico sobre imágenes 2D. Concretamente, las contribuciones de esta tesis son las siguientes: (1) Un sistema de reconocimiento de señales de tráfico basado en una red neuronal convolucional (Convolutional Neural Network) que incluye redes de transformadores espaciales (Spatial Transformer Networks), que establece un nuevo hito en el estado del arte superando los resultados y trabajos previamente publicados relacionados con el German Traffic Sign Recognition Benchmark (GTSRB) [69]. (2) Un estudio profundo de las capacidades de la red neuronal convolucional propuesta y del impacto en el rendimiento que tienen las capas de transformadores espaciales dentro de la red. (3) Análisis del efecto que tienen distintos algoritmos de optimización basados en gradientes descendentes sobre la red neuronal propuesta. (4) Evaluación de la red neuronal utilizando distintos conjuntos de datos públicos europeos de clasificación de señales de tráfico. (5) Evaluación del sistema de clasificación propuesto teniendo como entrada las imágenes detectadas a través del procesado de nubes de puntos 3D. (6) Presentación y estudio de algoritmos actuales de detección de objetos basados en redes neuronales convolucionales, como Faster R-CNN, R-FCN, SSD y YOLO. (7) Análisis y evaluación de detectores de objetos específicamente adaptados al problema de la detección de señales de tráfico sobre imágenes 2D. Dicha evaluación incluye métricas clave a la hora de tomar decisiones, como son la precisión media promedio (mAP), consumo de memoria, tiempo de ejecución, número de operaciones de punto flotante, número de parámetros de los modelos y el efecto que tienen el tamaño de las imágenes a la hora de realizar inferencias.

Estas contribuciones tienen aplicaciones prácticas reales, como por ejemplo en

coches autónomos, o en el inventariado automatizado y mantenimiento de la señalización en carreteras. La red neuronal convolucional de clasificación propuesta supera al sistema visual humano, su tiempo de inferencia es bajo y puede desplegarse como un servicio independiente que funciona en tiempo real. Por otro lado, el trabajo realizado sobre los detectores de señales de tráfico, permite a los lectores e investigadores elegir el modelo que mejor se adapte a las restricciones del entorno, siendo R-FCN Resnet 101 el detector que alcanza el mejor equilibrio entre precisión y tiempo de ejecución o inferencia, Faster R-CNN Inception Resnet V2 el que logra la mejor precisión, y SSD Mobilenet el que mejor se adapta a entornos móviles y dispositivos embebidos.

Toda la investigación de esta tesis doctoral ha sido parcialmente respaldada por el Ministerio de Economía y Competitividad de España a través de los proyectos "Hermes-Smart Citizen"(TIN2013-46801-C4-1-R) y "VICTORY"(TIN2017-82113-C2-1-R). Además, queremos dar las gracias a NVIDIA por la GPU Titan Xp donada a nuestro equipo de investigación para realizar este trabajo.

1.2. Metodología de investigación

Este trabajo sigue una técnica de investigación científica estándar [36] que incluye las siguientes fases:

1. **Definir el problema de investigación.** Los sistemas de detección y clasificación de señales de tráfico conforman una parte esencial coches autónomos, y mantenimiento e inventariado de la infraestructura de las carreteras, aplicándose en la mayoría de los casos sistemas con un alto coste económico. En esta tesis se propone un sistema de clasificación de señales de tráfico en tiempo real que supera al sistema visual humano y varios sistemas de detección de señales de tráfico que se pueden utilizar en distintos casos de uso, dependiendo de las restricciones del entorno.
2. **Revisión de la literatura.** Durante el periodo de esta tesis doctoral, se ha

realizado una búsqueda y lectura profunda sobre investigaciones basadas en sistemas de reconocimiento de señales de tráfico, tal como se muestra en las referencias incluidas en cada artículo.

3. **Formular hipótesis.** El grupo de investigación discutió cómo aplicar sistemas de reconocimiento de señales de tráfico basados en técnicas de inteligencia artificial, como pueden ser las redes neuronales convolucionales, con el fin de crear sistemas más precisos y eficientes.

4. **Diseño de la investigación.** Los sistemas de detección y clasificación fueron analizados para encontrar puntos claves susceptibles de ser modificados, como pueden ser distintas arquitecturas de redes convolucionales, o la inclusión de redes de transformadores espaciales en las arquitecturas de red neuronal propuestas previamente en el estado del arte.

5. **Recolectar datos.** Varios conjuntos de datos comúnmente utilizados y validados por la comunidad científica para tareas de reconocimiento de señales de tráfico fueron analizados y empleados durante esta tesis doctoral. Además, se generó un dataset de señales de tráfico españolas utilizando los sistemas de detección y clasificación descritos en los posteriores artículos.

6. **Ejecución del proyecto.** Las arquitecturas de redes neuronales diseñadas fueron entrenadas y validadas utilizando los datos anteriormente citados.

7. **Análisis de datos.** La información y resultados obtenidos por los sistemas desarrollados fueron analizados usando métricas estándar y comparados con resultados de investigaciones previas.

8. **Interpretar e informar.** Una vez que los sistemas propuestos han sido analizados y sus resultados interpretados, varios artículos de investigación fueron publicados como resultados de nuestras hipótesis.

1.3. Pregunta de investigación

La pregunta de investigación que conduce esta tesis doctoral es: *¿podemos mejorar la precisión de sistemas de reconocimiento de señales de tráfico sin disminuir la eficiencia en términos de requisitos de memoria utilizando técnicas englobadas en el ámbito de la inteligencia artificial?*

Dentro del contexto de la clasificación de señales de tráfico, nos centramos en desarrollar varias arquitecturas de red de neuronal que estuviesen compuestas tanto por capas convolucionales como por capas de transformadores espaciales. Los transformadores espaciales permiten realizar operaciones de transformaciones afines sobre las imágenes y los mapas de características, de modo que la red aprende a centrarse exclusivamente en la señal de tráfico, eliminado el fondo, realizando rotaciones, traslaciones, etc. El objetivo final fue analizar cómo estos elementos se comportaban al aplicar diferentes optimizadores basados en algoritmos de descenso de gradientes.

Dentro del contexto de la detección de señales de tráfico, nos centramos en analizar y comparar exhaustivamente el comportamiento de distintas redes neuronales propuestas en la literatura, adaptadas específicamente al contexto de la detección de señales.

1.4. Criterios de éxito

El éxito se logrará si la pregunta de investigación se resuelve. Esto significa que debemos comprobar, por un lado, que el sistema de clasificación es capaz de categorizar la señal de tráfico dada como entrada (stop, prohibido el paso, límite de velocidad 50, etc.) en imágenes de escenarios reales. Por otro lado, que el sistema de detección es válido para localizar dónde se encuentran las señales de tráfico dada una imagen de un escenario real como puede ser una carretera de autovía o una urbana. Los resultados mostrados en esta tesis en forma de artículos de investigación demuestran que coinciden con la predicción formulada en la hipótesis de partida.

En el primer caso, esta tesis establece un nuevo récord de precisión en el German Traffic Sign Recognition Benchmark, superando incluso al sistema visual humano. En el segundo caso, el análisis y comparación exhaustiva de ocho detectores de señales de tráfico basados en aprendizaje profundo, nos permite ofrecer modelos preparados para ser usados en entornos reales, así como ayudar a los lectores a elegir el mejor modelo que se adapte a sus necesidades en términos de precisión, tiempo de ejecución, consumo de memoria, etc.

1.5. Propiedades analizadas y discutidas

La detección y clasificación de señales de tráfico se abordan desde varios puntos de vista:

- **Precisión.** En relación a la precisión de los modelos de redes neuronales entrenados y evaluados utilizando conjuntos de datos públicos de reconocimiento de señales de tráfico. Esta propiedad es importante para que el sistema sea robusto.
- **Eficiencia.** En relación al coste computacional, consumo de memoria, y tiempo de ejecución, entre otros, que tienen dichos modelos. Esta propiedad permite seleccionar el modelo que mejor se adapte a las necesidades del lector.

El trabajo presentado en este documento propone una solución para mejorar la robustez de los sistemas de reconocimiento de señales de tráfico, al mismo tiempo que se mejora la eficiencia.

1.6. Esquema de la tesis

Este documento está estructurado de la siguiente forma. En la Parte I se presenta la introducción. En la Parte II se muestran los tres artículos de investigación

derivados de esta tesis doctoral, divididos en 3 capítulos: Capítulo 2 - "Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems"; Capítulo 3 - "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods"; Capítulo 4 - "Evaluation of deep neural networks for traffic sign detection systems". Las revistas donde se han publicado estos artículos están incluidas en el ranking JCR de Thomson Reuters y todos ellos están relacionados con el problema de la detección y clasificación de señales de tráfico:

- **Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems**. Álvaro Arcos-García, Mario Soilán, Juan A. Álvarez-García, Belén Riveiro. Publicado en Expert Systems with Applications, Elsevier, ISSN: 0957-4174, Fecha de Publicación: Diciembre 2017, Volumen: 89, En Páginas: 286-295, DOI: `https://doi.org/10.1016/j.eswa.2017.07.042`, [JCR-2017 3.768] [Q1 en Computer Science, Artificial Intelligence (20/132)].

- **Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods**. Álvaro Arcos-García, Juan A.Álvarez-García, Luis M.Soria-Morillo. Publicado en Neural Networks, Elsevier, ISSN: 0893-6080, Fecha de Publicación: Marzo 2018, Volumen: 99, En Páginas: 158-165, DOI: `https://doi.org/10.1016/j.neunet.2018.01.005`, [JCR-2017 7.197] [Q1 en Computer Science, Artificial Intelligence (7/132)].

- **Evaluation of deep neural networks for traffic sign detection systems**. Álvaro Arcos-García, Juan A.Álvarez-García, Luis M.Soria-Morillo. Publicado en Neurocomputing, Elsevier, ISSN: 0925-2312, Fecha de Publicación: Noviembre 2018, Volumen: 316, En Páginas: 332-344, DOI: `https://doi.org/10.1016/j.neucom.2018.08.009`, [JCR-2017 3.241] [Q1 en Computer Science, Artificial Intelligence (27/132)].

Un resumen del ranking de estos artículos de investigación se puede encontrar en la Tabla 1.1.

Título	Revista	F.I.	Ranking
Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems	Expert Systems with Applications 2017	3.768	Q1
Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods	Neural Networks 2018	7.197	Q1
Evaluation of deep neural networks for traffic sign detection systems	Neurocomputing 2018	3.241	Q1

Tabla 1.1: Resumen de artículos publicados en revistas JCR indexadas.

Finalmente, en la Parte III, se exponen comentarios finales, conclusiones y se discute el trabajo futuro.

PARTE II

Trabajos de investigación seleccionados

CAPÍTULO 2

EXPLOITING SYNERGIES OF MOBILE MAPPING SENSORS AND DEEP LEARNING FOR TRAFFIC SIGN RECOGNITION SYSTEMS

Resumen

Este trabajo nace de una colaboración con investigadores del grupo de Geotecnologías Aplicadas de la Universidad de Vigo en el contexto del proyecto "Healthy and Efficient Routes in Massive Open-Data Based Smart Cities-Citizen"(TIN2013-46801-C4-1-R), financiado por el Ministerio de Econonomía y Competitividad de España. Los miembros de dicho grupo de investigación son expertos en el uso de Mobile Mapping Systems (MMS) en sistemas de mantenimiento de inventario de la vía pública.

El artículo de investigación presenta un sistema eficiente de reconocimiento de señales de tráfico divido en dos fases. En primer lugar, los datos de nubes de puntos 3D se adquieren mediante un sistema LINX Mobile Mapper y se procesan para detectar automáticamente las señales de tráfico basándose en el material reflectivo

que contienen. En segundo lugar, la clasificación de las señales se lleva a cabo sobre la proyección de la nube de puntos en imágenes RGB, aplicando una red neuronal profunda que contiene capas de transformadores espaciales y convolucionales. Esta red se evalúa utilizando tres conjuntos de datos de señales de tráfico europeas. En el German Traffic Sign Recognition Benchmark (GTSRB), la red propuesta supera a los trabajos publicados previamente y logra el primer puesto del ranking con una precisión del 99,71 %. Además, se genera y publica un nuevo conjunto de imágenes de señales de tráfico españolas que puede ser utilizado en futuras tareas de clasificación.

Expert Systems With Applications 89 (2017) 286–295

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems

Álvaro Arcos-García[a,*], Mario Soilán[b], Juan A. Álvarez-García[a], Belén Riveiro[b]

[a] Computer Languages and Systems Department, University of Seville, Avda. Reina Mercedes s/n, Seville 41012, Spain
[b] Department of Materials Engineering, Applied Mechanics & Construction, University of Vigo, Torrecedeira 86, Vigo 36208, Spain

ARTICLE INFO

Article history:
Received 12 May 2017
Revised 4 July 2017
Accepted 25 July 2017
Available online 26 July 2017

Keywords:
Mobile mapping sensors
Point cloud
Traffic sign
Deep learning
Convolutional neural network
Spatial transformer network

ABSTRACT

This paper presents an efficient two-stage traffic sign recognition system. First, 3D point cloud data is acquired by a LINX Mobile Mapper system and processed to automatically detect traffic signs based on their retro-reflective material. Then, classification is carried out over the point cloud projection on RGB images applying a Deep Neural Network which comprises convolutional and spatial transformer layers. This network is evaluated in three European traffic sign datasets. On the GTSRB, it outperforms previous state-of-the-art published works and achieves top-1 rank with an accuracy of 99.71%. Furthermore, a Spanish traffic sign recognition dataset is released.

1. Introduction

According to the European Union Road Federation (ERF), there exists a negative trend regarding road infrastructure investments and maintenance, as the funding for those expenses is decreasing since 2008 (European Union Road Federation, 2015). This report points out that this negative trend has a massive economic impact in the mid and long term, as both the investments required for the maintenance of the infrastructure and the vehicle operating costs increase exponentially as the condition of the road deteriorates. Vertical signs are an essential asset which regulate the traffic and guide road users. Traffic signs need to be visible during both day and night time, therefore periodic inspections should ensure the visual performance of the sign. However, the ERF pointed out the existence of an alarming backlog in traffic sign maintenance in many European countries because it reduces the safety of the roads as traffic signs might have faded colors or lose their retro-reflective properties. Given that accidents caused by infrastructure deficiencies result in high human and economic costs, investing in road infrastructure (and specifically in vertical signage) will have a positive impact in terms of road safety and economic return. There are different strategies for the maintenance and replacement of traffic signs. They can be replaced in fixed time intervals, or periodic inventories can be established. Typically, these inventories are carried out manually and in situ. Nowadays, remote-sensing technology allows the road to be measured faster, safer and expending less resources, hence significantly improving the outcomes of investments in road infrastructures. Mobile Mapping Systems (MMS) are able to collect large amounts of 3D and 2D data using Mobile Laser Scanner (MLS) technology together with imagery systems. The 3D representations of surveyed environments are dense and accurate and provide reliable information about the geometric and radiometric properties of the scanned areas (Puente, González-Jorge, Martínez-Sánchez & Arias, 2013a). However, despite the increasing attention this technology is receiving, there exist some limitations given by the resolution of the scanning system and the storage and processing capabilities of the computers. For that reason, imagery data may be useful for some applications. Classifying 2D images of traffic signs captured by RGB sensors is a traditional research topic in computer vision since developing a robust traffic sign recognition system is still a challenging task.

This research is motivated by (1) the need to develop methodologies allowing for the automation of road infrastructure inspection activities and therefore improving inventory and maintenance of a huge financial public asset as it is the road network, and (2) the potential usefulness of combining different data sources from a Mobile Mapping System, complementing an accurate 3D description of the road network with RGB imagery, in order to offer precise semantic descriptions.

* Corresponding author.
E-mail addresses: aarcos1@us.es (Á. Arcos-García), msoilan@uvigo.es (M. Soilán), jaalvarez@us.es (J.A. Álvarez-García), belenriveiro@uvigo.es (B. Riveiro).

http://dx.doi.org/10.1016/j.eswa.2017.07.042
0957-4174/

A robust pipeline is proposed to efficiently process LiDAR data, detect with high accuracy vertical traffic signs and recognize their classes applying a Deep Neural Network (DNN) to the corresponding 2D images. The growing acceptance in developed countries of the benefits of LiDAR implies several countries can apply this robust methodology.

The rest of the paper is organized as follows. Section 2 analyses the state of the art of traffic sign recognition systems from two points of view, LiDAR and 2D images. Section 3 shows the proposed methodology and results are explained in Section 4. Finally conclusions are drawn in Section 5.

2. Related works

Traffic sign recognition systems (TSRS) are helpful for Advanced Driver Assistance Systems (ADAS) or autonomous vehicles, nevertheless, a wide range of challenges needs to be overcome such as changing ambient lighting conditions, occlusions, focusing or blurring problems and deterioration or deformations due to ageing or vandalism. Furthermore, the variety of different traffic signs that have to be distinguished is very wide and diverse for different countries. For example, there are more than 200 traffic sign classes in Spain (Spanish Government, 2003), Germany[1] and Belgium.[2] All of these issues affect TSRS and are important factors that should be considered.

One of the main problems before the year 2011 was the lack of a public traffic sign dataset. The Belgian Traffic Sign Classification dataset (BTSC) (Timofte, Zimmermann, & Van Gool, 2011) and the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp, Schlipsing, Salmen, & Igel, 2011), a multi-category classification competition, solved this issue and boosted the research in TSRS. GTSRB made publicly available a traffic sign dataset with more than 50,000 labeled samples divided into 43 classes. It is commonly used to evaluate the performance of computer vision algorithms and compare them versus the human visual system (Stallkamp, Schlipsing, Salmen, & Igel, 2012).

Mathias, Timofte, Benenson, and Van Gool (2013) propose fine grained classification applying different methods through a pipeline of three stages: feature extraction, dimensionality reduction and classification. On GTSRB, they reach 98.53% of accuracy merging grayscale values of traffic sign images and Histogram of Oriented Gradients (HOG) based features, reducing the dimensionality through Iterative Nearest Neighbors-based Linear Projections (INNLP) and classifying with Iterative Nearest Neighbors (INN). Although Support Vector Machines (SVMs) (Maldonado-Bascón, Acevedo-Rodríguez, Lafuente-Arroyo, Fernández-Caballero, & López-Ferreras, 2010), Random Forests (Zaklouta, Stanciulescu, & Hamdoun, 2011) and Nearest Neighbors (Gudigar, Chokkadi, Raghavendra, & Acharya, 2017) classifiers have been used to recognize traffic sign images, Convolutional Neural Networks (ConvNets or CNNs) (Lecun, Bottou, Bengio, & Haffner, 1998) showed particularly high classification accuracies in the competition. Cireşan, Meier, Masci, and Schmidhuber (2012) won the GTSRB contest with a 99.46% accuracy thanks to a committee of 25 ConvNets with 3 convolutional layers and 2 fully connected layers each. Sermanet and LeCun (2011) use multi-scale ConvNets achieving an accuracy of 98.31% and second place in the GTSRB challenge. In 2014, Jin, Fu, and Zhang (2014) proposed a hinge loss stochastic gradient descent method to train ConvNets that brought off 99.65% accuracy and offered a faster and more stable convergence than previous works.

Most TSRS rely exclusively on image or video processing, for instance, Kaplan Berkaya, Gunduz, Ozsen, Akinlar, and Gunal (2016) propose a circle detection algorithm along with an RGB-based color thresholding procedure during detection stage over 2D images which are classified applying an ensemble of features comprising HOG, Gabor and local binary patterns (LBP) within a SVM afterward. Nevertheless, the use of MMS allows new approaches. A MMS is formed by different components, namely mapping sensors (typically laser scanners and RGB or infrared cameras), a navigation unit which is composed of Global Navigation Satellite Systems, Inertial Measuring Units and Distance Measurement Indicators, and a time referencing unit which allows the temporal synchronization of the different measurements collected. In recent years, a large number of methodologies have been developed which automatically process the geometric and radiometric information acquired by a MMS for different applications. Among them, object detection and recognition is a topic that has received considerable attention in the literature. Oliveira, Nunes, Peixoto, Silva, and Moita (2010) propose the semantic fusion of point cloud data gathered with laser scanners and computer vision to detect pedestrians in urban scenarios.

With regard to traffic signs, Pu, Rutzinger, Vosselman, and Elberink (2011) classify planar shapes in point clouds using geometric based approaches. González-Jorge, Riveiro, Armesto, and Arias (2013) show that laser scanner systems can capture the geometry of traffic sign panels based on the intensity values of those laser beams that are reflected on the panels. These values are much higher than those in their surroundings, owing to the retroreflective properties of traffic signs paint. Riveiro, Díaz-Vilarino, Conde-Carnero, Soilán, and Arias (2016) rely on the intensity attribute of the point clouds in order to segment reflective elements. Then, they recognize the shape of the detected elements by evaluating their contour and fitting a polynomial curve to it, which is compared with a set of patterns that represent simple shapes. However, this approach faced some limitations; distinguishing between circular shapes and octagonal shapes was not possible due to the low resolution of the point cloud, and the specific meaning of a traffic sign could not be retrieved. Recently, some work has been published which combines 3D point cloud information and imagery data. Wen et al. (2016) detect traffic signs on a preprocessed point cloud using a single threshold value and implement an on-image sign detection which consist on the projection of detected signs on 2D images and a classification by means of SVM using a combination of Hue SIFT and HOG feature vectors. Yu et al. (2016) present a similar approach which uses a bag of visual phrases for the detection and a deep Boltzmann machine hierarchical classifier, which is a deep learning model that allows to generate highly distinctive features.

3. Methodology

In this work we propose the next methodology: initially our vehicle equipped with LiDAR and RGB cameras gathers information (3D point cloud and 2D imagery). Then, the point cloud is processed to automatically detect traffic signs based on their retroreflective properties. Furthermore, each detected traffic sign is associated with its respective RGB images. Finally, a DNN is applied to classify the type of traffic sign from the filtered set of RGB images (see Fig. 1).

The next subsections detail the traffic sign detection, point cloud projection on RGB images and traffic sign classification.

3.1. Traffic sign detection from 3D point clouds

This subsection summarizes the traffic sign detection method. It is based on Soilán, Riveiro, Martínez-Sánchez, and Arias (2016)

[1] https://www.adac.de/_mmm/pdf/fi_verkehrszeichen_engl_infobr_0915_30482.pdf (accessed 17.03.22).

[2] http://wiki.openstreetmap.org/wiki/Road_signs_in_Belgium (accessed 17.03.22).

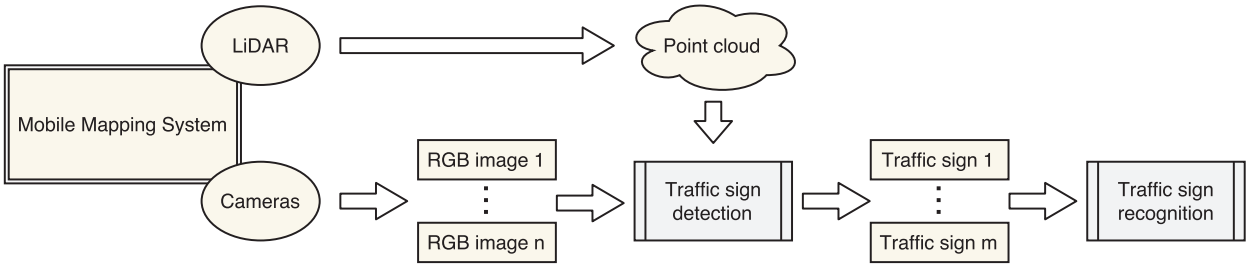


Fig. 1. Proposed methodology. Traffic sign detection by means of LiDAR data processing and traffic sign recognition through a DNN.

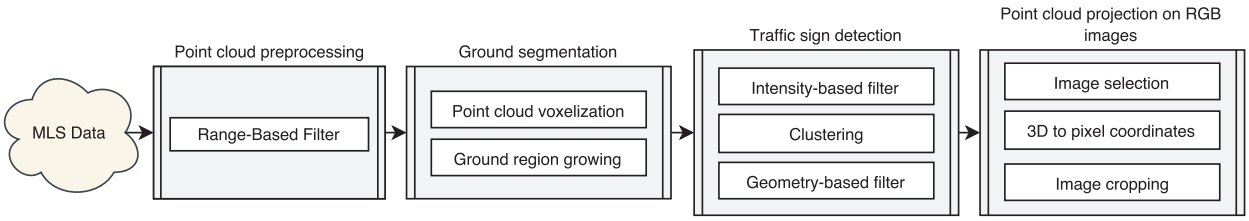


Fig. 2. Point cloud processing. Workflow of the point cloud processing methodology.

work and consists of a sequence of data processing modules which aim to detect traffic sign panels in 3D point clouds acquired by a MMS. The global processing chain can be seen in Fig. 2.

3.1.1. Point cloud preprocessing

In order to reduce the amount of data processed, redundant or unnecessary information should be removed from the input point cloud. For that purpose, the distance from the 3D point cloud points to the trajectory registered by the MMS is computed. Once all the distances are computed, points further than 15 m from the trajectory are filtered out, as the objects to be studied are supposed to be displayed alongside the road.

3.1.2. Ground segmentation

Next step of the method consists of the segmentation of the ground. Let $P = (x, y, z, I, t)$ be a 3D point cloud acquired by a MMS, where (x, y, z) are the 3D coordinates of the point cloud, I is the intensity of the returned pulse for each measured 3D point, and t is the time stamp of each point. Let $T = (x_r, y_r, z_r, t_r)$ be the trajectory of the MMS during the acquisition of the point cloud P, as measured by the positioning system of the vehicle.

Here, the input point cloud P is voxelized, that is, a $N_x \times N_y \times N_z$ cubic grid with size g_s is defined such that a voxel with a coordinate (x_i^v, y_i^v, z_i^v) within the grid and a voxel index is assigned to every point (x_i, y_i, z_i) in according to Eqs. (1)–(4).

$$x_i^v = round(x_i - min(x))/g_s \tag{1}$$

$$y_i^v = round(y_i - min(y))/g_s \tag{2}$$

$$z_i^v = round(z_i - min(z))/g_s \tag{3}$$

$$id_i^v = (z_i - min(z))/g_s \tag{4}$$

Let $V(P) = (x, y, z, \mu_z, v_z)$ be the voxelized point cloud of P, and $V(P, id^v) = (x, y, z, \mu_z, v_z)$ be the voxel with index id^v, where (x, y, z) is the centroid, and (μ_z, v_z) are the vertical mean and variance, of the points in P with index id^v.

At this point, the ground segmentation is conducted based on a modification of Douillard et al. (2011) method for the partition of the ground. They cluster together adjacent voxels whose vertical mean and variance differences are less than certain thresholds, and select the largest partition as the ground. Here, voxels that belong to the ground are selected as seeds for a region growing process where vertical mean and variance differences between adjacent voxels are used as criteria to decide whether a voxel belongs to the ground or not.

The ground seeds are selected using the trajectory T and the fact that the mapping system always travels over the ground. A K-Nearest-Neighbor algorithm is used to obtain the closest voxel for each point in the trajectory such that the elevation of the voxel is smaller than the elevation of the trajectory. That way, a set of voxels in the ground is obtained, making the region growing process faster and eliminating the necessity of clustering and selecting the largest region.

This process is driven by two parameters, which are the thresholds for vertical mean and vertical variance differences, d_μ and d_σ. This method aims for a coarse segmentation of the ground, including curbs and speed bumps. The parameters have been empirically tuned, and for the study case experiments their values are $d_\mu = 0.1m$ and $d_\sigma = 0.05$.

3.1.3. Detection of traffic signs based on the intensity data

Let $P_{ng} \subset P$ be the non-ground segment point cloud (Fig. 3a), which is obtained after filtering out the ground segment from the point cloud.

Traffic signs are panels made of retro-reflective materials. Therefore, the intensity property of the point cloud, which is directly related with the reflectance of the objects can be used for the detection of traffic signs. It can be assumed that the intensity distribution of both reflective and non-reflective points in P_{ng} follows a normal distribution (Riveiro et al., 2016). Therefore, an unsupervised classification algorithm based on Gaussian Mixture Models (GMM) is proposed. GMM are multivariate distributions consisting of one or more Gaussian distribution components. Here, a mixture distribution with two components is estimated given the intensity values of the points in P_{ng}. Then, each point in the cloud is assigned to one of the components, and those points assigned to the component with largest mean are selected for the next processing step.

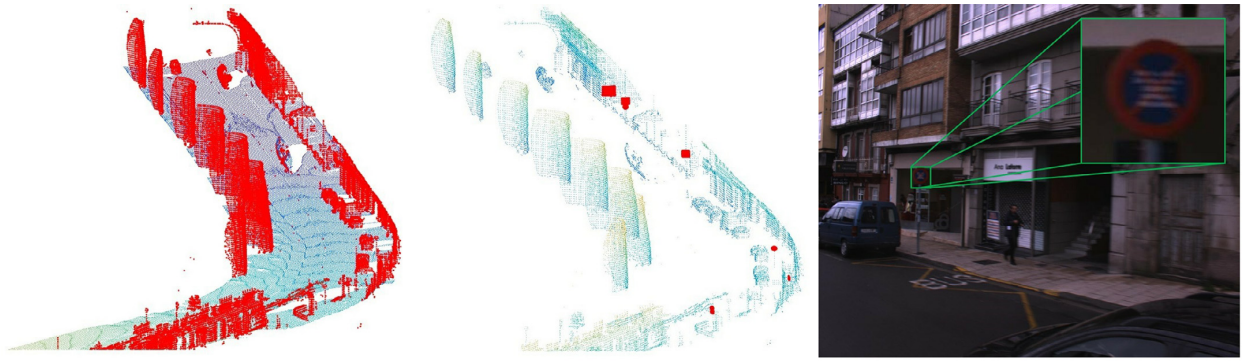


Fig. 3. Traffic sign detection. (a) The ground segment is filtered out from the point cloud. Therefore, only non-ground points (colored in red) are analyzed in the subsequent steps. (b) Both intensity and geometry filters are applied in order to segment traffic sign panels (colored in red). (c) The 3D point cloud traffic sign panels are projected on 2D images and the bounding box of the projection is used for cropping the images, facilitating the traffic sign recognition process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The selected points have large intensity values, but they are still unorganized, that is, there is no relation between the points in the cloud. Hence, a clustering algorithm is applied in order to group together points that may belong to the same object. DBSCAN algorithm (Ester, Kriegel, Sander, & Xu, 1996) groups points which are closely packed together while selecting isolated points as outliers. This algorithm allows to group points that belong to different objects in a set of clusters $C = \{C_1, \ldots, C_i, \ldots, C_n\} | C_i \subset P_{ng}$. That is, each cluster C_i contains a group of points from P_{ng} which belong to the same object and have large intensity values.

Finally, C is filtered using the knowledge about the geometry of the traffic sign panels, that is, they are planar surfaces, and they have an enclosed range of heights. First, the dimensionality of each cluster is analyzed. For each $C_i \subset P_{ng}$ a Principal Component Analysis (PCA) of the covariance matrix of the points within the cluster is carried out such that the planarity of C_i is according to Eq. (5), where λ_i is the $i-th$ eigenvalue returned by PCA.

$$a_{2D} = \left(\sqrt{\lambda_2} - \sqrt{\lambda_3}\right) \Big/ \sqrt{\lambda_1} \tag{5}$$

If $a_{2D} < 1/3$, the cluster cannot be labeled as a plane (Gressin, Mallet, Demantké, & David, 2013) and therefore it is filtered out. Subsequently, a height filter is applied such that clusters with heights smaller than 25cm are also filtered out. Both filters eliminate objects with reflective properties which are not planar or small, such as vehicle license plates. The detection process outputs a subset of C, $C_{ts} \subset C$ which contain traffic sign panels (Fig. 3b).

3.2. Point cloud projection on RGB images

The resolution of traffic sign panel clusters C_{ts} is not enough to obtain semantic information of the traffic sign. Although it is possible to recognize different shapes, most of the visual information is lost in the 3D point cloud. Therefore, the traffic sign recognition task is carried out using RGB images taken by four cameras installed in the MMS. The camera calibration parameters, namely radial distortion parameters (k_1, k_2), focal length (f_j, $j = 1 \ldots 4$), pixel size (s_{pix}), and pixel coordinates of the principal point (c_x, c_y) are known, together with the orientation parameters that relate the camera coordinate system and the vehicle (Puente, González-Jorge, Riveiro, & Arias, 2013b). Moreover, the position of the vehicle and the time stamp is known for each RGB image. For each cluster $C_i \subset C$, the average time stamp t_{ave} of the 3D points is computed and only those images whose time stamp is in the interval $t_{ave} \pm 5s$ are analyzed. Let p_{ih} be 3D homogeneous coordinates of the points of the traffic sign panel i. First, the coordinates are transformed from the global coordinate system to the vehicle coordinate system following (Eq. (6)):

$$p_{ih}^{c} = (T_{ab} T_{ac})^{-1} p_{ih}^{A} \tag{6}$$

Where A is the global coordinate system, B is the GNSS coordinate system, C is the vehicle coordinate system, and T_{ab}, T_{ac} are the transformation matrices between AB and BC.

Once the traffic sign panel coordinates and the camera position are both related to the vehicle coordinate system, the 3D points can be projected onto the plane of each camera and the coordinates with respect to the camera frame (d_u, d_v) can be obtained. A radial distortion model is applied to correct the coordinates (tangential distortion is not considered), and pixel coordinates can be retrieved using the pixel size value together with the coordinates of the principal point (Eqs. (7) and (8)).

$$x_{pix} = d_u (k_1 r^2 + k_2 r^4) + c_x / s_{pix} \tag{7}$$

$$y_{pix} = d_v (k_1 r^2 + k_2 r^4) + c_y / s_{pix} \tag{8}$$

Once every point of a traffic sign panel is projected into an image, the bounding box of the pixel coordinates is retrieved. The image is automatically cropped according to the bounding box with a margin of a 25% (Fig. 3c) to compensate for possible calibration errors and add some background for training classification models.

3.3. Traffic sign recognition

Once the RGB images have been selected and the image samples containing the traffic signs have been stored, the classification process starts. As seen in Section 2, ConvNets have been widely used to classify traffic signs. In our work a traffic sign recognition system based on DNN is proposed, whose main blocks are convolutional and spatial transformer layers. In the following subsections, the initial dataset, the data preprocessing and our DNN architecture are described.

3.3.1. Initial dataset preparation

In Spain there is not any public dataset available for its 252 traffic sign categories. Gathering a sufficient number of images of all the categories is a challenging task. In this work, an initial dataset with 83 classes has been obtained thanks to the filtered images collected with the MLS explained above, combined with images from the German and Belgian dataset that are similar to

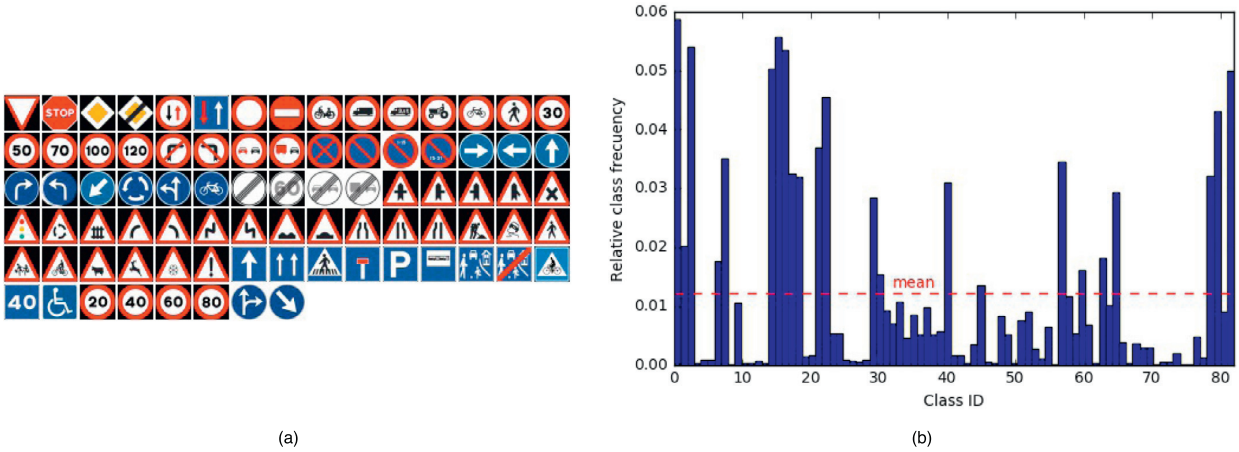


Fig. 4. Mixset dataset. (a) Traffic sign categories. (b) Relative class frequencies.

Table 1
European datasets mixed.

Dataset	Training images	Validation images	Classes
GTSRB	39,209	12,630	43
Adapted BTSC	4024	2263	58
Spain	897	452	43
Mixset	44,130	15,345	83

Spanish case. The dataset is available at https://daus-lab.github.io/spanish-traffic-sign-dataset.

All the collected images from Spain were manually classified in a collaborative way through a web site designed specifically for that task. Only those categories with more than six examples were used in the initial dataset. Later, images are randomly mixed and split into training and validation sets five times in order to evaluate the recognition system through cross-validation. Each of these folds is composed by 897 training images and 452 validation images distributed in 43 categories. As may be seen, the scale of the collected dataset is small and will be enlarged in future work even though the current dataset version along with the Mixset ground truth files will be kept for reproducibility and comparability purposes.

In the German traffic sign recognition dataset, the training set has 39,209 images and validation set consists of 12,630 that are used to measure the performance of algorithms in the GTSRB (Stallkamp et al., 2011). All the German categories are included in the Spanish Road Traffic Regulations document (Spanish Government, 2003).

The Belgian traffic sign classification dataset was carefully revised because it contains categories that cluster different traffic signs types (e.g. 50 speed limit sign and 70 speed limit sign). It also includes some classes that were removed because they are not defined in the Spanish Road Traffic Regulations document. Thus, testing images from Belgian dataset were used as validation set. Some empty categories were filled selecting one random sample per each road track from training set and moving it to our validation set, according to Sermanet and LeCun (2011). After adaptation, the Belgian dataset consists of 4024 training images and 2263 validation images divided into 58 categories.

Classes of the three datasets were related to each other, resulting in an initial dataset (Table 1) of 44,130 training images, 15,345 validation images and 83 traffic sign types (Fig. 4a). The usage of the Spanish dataset permits to add 13 unique traffic sign categories that were not in the German or Belgian ones. From now on, we will refer to this dataset as Mixset. Note that Mixset is highly imbalanced, for example, 9 out of 83 categories in training set and 21 out of 83 classes in validation set have less than 10 samples. By contrast, 17 out of 83 types of traffic signs contain more than 1000 training samples (Fig. 4b).

3.3.2. Data pre-processing of Mixset images

Mixset samples are raw RGB and sizes vary from 21×22 to 700×700 pixels. All of them are up-sampled or down-sampled to 48x48 pixels and preprocessed with global and local contrast normalization with Gaussians kernels (Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009) that centers each input image around its mean value and enhances edges.

3.3.3. Deep Neural Network architecture

The proposed method to recognize traffic signs is a DNN that combines several convolutional, spatial transformer, non-linearity, contrast normalization and max-pooling layers. It acts as a feature extractor that maps raw pixel information of the input image into a tensor to be classified by two fully connected layers. Spatial transformer layers are used to perform explicit geometric transformations on input images and feature maps in order to focus on the object to be learned, removing progressively background and geometric noise. All variable parameters used in each of these layers are optimized together through minimization of the misclassification error over the Mixset training set.

The convolutional layers carry out a 2-dimensional convolution of its $n-1$ input maps with a filter of size $F_x^n \times F_y^n$, where x and y represent the size of each dimension. Each convolutional layer is composed by neurons which have learnable biases and weights. During the feed forward process of the neural network, each filter is convolved across the height and width of the input map, performing a dot product that produces a 2-dimensional activation map of that filter. The resulting activations of the n output maps are given by the sum of the $n-1$ convolutional responses that are passed through a non-linear activation function f where n is the convolutional layer, i and j represent the input map and the output map respectively, a indicates a map of size $x \times y$, the weights w_{ij} are represented as a filter of size $F_x \times F_y$ which connects the input map with the output map, and b_j is the bias of the output map (Eq. (9)). Rectified Linear Units (ReLU) layers are used to compute

Fig. 5. Spatial transformer network. Input images on the first row and computed output images on the second row.

the non-linear activation function.

$$a_j^n = \sum_{i=1}^{n-1} a_i^{n-1} * w_{ij}^n + b_j^n \tag{9}$$

ReLU layers are made up of neurons that apply the activation function $f(x) = max(0, x)$, where x is the input to a neuron. It enhances the non-linear properties of the network, including the decision function, without affecting the learnable parameters of the convolutional layer.

Max-pooling layers are used to reduce progressively the spatial size of the representation, in order to decrease the amount of parameters, computation in the network and to control overfitting by selecting superior invariant features, and improving generalization. The output of this layer is given by the maximum activation over non-overlapping regions of filter size $F_x \times F_y$, where the input map is downsampled by a factor of F_x and F_y along both width and height, nevertheless depth dimension remains unchanged.

Contrast normalization layers (Jarrett et al., 2009) are used to normalize the contrast of an input map through subtractive local normalization and divisive local normalization. Both operations use a Gaussian kernel, and are computed at local spatial regions of the input map on a per feature basis.

Fully connected layer neurons have full connections to all activations in the previous layer, in other words, it combines the outputs of the previous layer into a 1-dimensional feature vector. The last fully-connected layer of the network performs the classification task since they have one output neuron per class, followed by a logarithmic soft-max activation function.

Spatial Transformer Networks (Jaderberg, Simonyan, Zisserman, Kavukcuoglu, 2015) aim to perform geometric transformation on an input map so that provides to ConvNets the ability to be spatially invariant to the input data in a computationally efficient manner. Thanks to such transformations, there is no need for extra training supervision, handcrafted data augmentation (e.g. rotation, translation, scale, skew, cropping) or dataset normalization techniques. This differentiable module can be inserted into existing convolutional architectures since the parameters of the transformation that are applied to feature maps are learned by means of a backpropagation algorithm. Spatial transformer networks consist of 3 elements: the localization network, the grid generator and the sampler (Fig. 6).

The localization network $f_{loc}()$ takes an input feature map $U \in R^{H \times W \times C}$, where H, W and C are the height, width and channels respectively, and outputs the parameters θ of the transformation T_θ to be applied to the feature map $\theta = f_{loc}(U)$. The dimension of θ depends on the transformation type T_θ that is being parameterized, being 6-dimensional in our proposed net since it performs a 2D affine transformation A_θ which allows translation, cropping, rotation, scale and skew. The localization network can comprise any number of convolutional and fully connected layers, and must have at least one final regression layer to generate the transformation parameters θ. Such parameters are used by the grid generator to create a sampling grid, which is a set of points where the input map has to be sampled to obtain the desired transformed output.

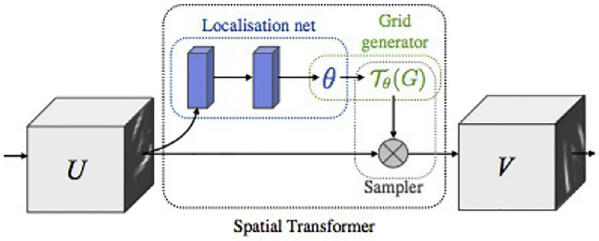


Fig. 6. Spatial transformer network components (Jaderberg et al., 2015).

Finally, the sampler uses as inputs the sampling grid and the input feature map U in order to perform a bilinear sampling which produces the transformed output feature map $V \in R^{H' \times W' \times C}$, where H', W' are the height and width of the sampling grid.

For source coordinates in the input feature map (x_i^s, y_i^s) and a learned 2D affine transformation matrix A_θ, the target coordinates of the regular grid in the output feature map (x_i^t, y_i^t) are given as follows (Eq. (10)):

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{10}$$

Regarding traffic sign recognition, spatial transformer networks learn to focus on the traffic sign removing gradually geometric noise and background so that only the interesting zones of the input are forwarded to the next layers of the network (Fig. 5). Up to our knowledge, no peer review work has been published including the spatial transformer unit into a ConvNet for the traffic sign recognition task.

Our proposed DNN consists of three main blocks that act as feature extractors and comprises a spatial transformer network, a convolutional layer, a ReLU layer, a max-pooling layer and a local contrast normalization layer. Then, the classification is carried out by two fully-connected layers separated by a ReLU layer. The last fully-connected layer is made of 83 neurons corresponding to each the traffic sign categories to be classified (Fig. 7).

The localization network of the three spatial transformer networks is built with a max-pooling layer followed by two blocks of convolutional, ReLU and max-pooling layers. Also in this case, the classification stage has 2 fully-connected layers and a ReLU one although the last fully-connected only contains 6 neurons that correspond to the parameters of the affine transformation matrix.

The DNN architecture proposed is shown in Tables 2 and 3. Convolutional layers stride is set to 1 in order to leave all spatial down-sampling computation to max-pooling layers, and zero padding is set to 2, in contrast with max-pooling layers, whose stride is set to 2 and zero padding to 0. The total parameters learned (weights) by this single DNN is 14,629,801 which is much less than in other ConvNets proposed for traffic sign recognition

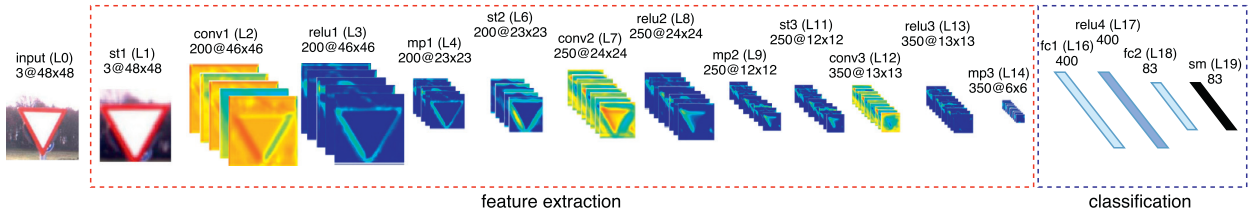


Fig. 7. DNN for traffic sign recognition proposed. Local contrast normalization layers have been omitted in the figure above to simplify its visualization as well as localization networks of spatial transformers. The *st* layers refer to spatial transformer networks, *conv* to convolutional layers, *mp* to max-pooling layers, *fc* to fully-connected layers and *sm* to soft-max layer.

Table 2
Detailed DNN architecture proposed for traffic sign recognition.

Layer	Type	# Maps and neurons	Kernel	# Weights
0	Input	3 m. of 48 × 48 n.		
1	Spatial Transformer 1	3 m. of 48 × 48 n.		3,833,506
2	Convolutional	200 m. of 46 × 46 n.	7 × 7	29,600
3	Non-linearity (ReLU)	200 m. of 46 × 46 n.		
4	Max-Pooling	200 m. of 23 × 23 n.	2 × 2	
5	Contrast Norm.	200 m. of 23 × 23 n.		
6	Spatial Transformer 2	200 m. of 23 × 23 n.		1,742,456
7	Convolutional	250 m. of 24 × 24 n.	4 × 4	800,250
8	Non-linearity (ReLU)	250 m. of 24 × 24 n.		
9	Max-Pooling	250 m. of 12 × 12 n.	2 × 2	
10	Contrast Norm.	250 m. of 12 × 12 n.		
11	Spatial Transformer 3	250 m. of 12 × 12 n.		1,749,956
12	Convolutional	350 m. of 13 × 13 n.	4 × 4	1,400,350
13	Non-linearity (ReLU)	350 m. of 13 × 13 n.		
14	Max-Pooling	350 m. of 6 × 6 n.	2 × 2	
15	Contrast Norm.	350 m. of 6 × 6 n.		
16	Fully connected	400 neurons	1 × 1	5,040,400
17	Non-linearity (ReLU)	400 neurons		
18	Fully connected	83 neurons	1 × 1	33,283
19	Soft-max	83 neurons		

Table 3
Localization network details of spatial transformers used in the main DNN. Kernel size of convolutional layers is set to 5 × 5 and max-pooling layers to 2 × 2. The annotation shown in the table is simplified, for instance, 3 of 48 × 48 stands for 3 maps of 48 × 48 neurons each one.

Layer/Type	Loc. net of ST 1	Loc. net of ST 2	Loc. net of ST 3
0/Input	3 of 48 × 48	200 of 23 × 23	250 of 12 × 12
1/Max-Pool	3 of 24 × 24	200 of 11 × 11	250 of 6 × 6
2/Conv	250 of 24 × 24	150 of 11 × 11	150 of 6 × 6
3/ReLU	250 of 24 × 24	150 of 11 × 11	150 of 6 × 6
4/Max-Pool	250 of 12 × 12	150 of 5 × 5	150 of 3 × 3
5/Conv	250 of 12 × 12	200 of 5 × 5	200 of 3 × 3
6/ReLU	250 of 12 × 12	200 of 5 × 5	200 of 3 × 3
7/Max-Pool	250 of 6 × 6	200 of 2 × 2	200 of 1 × 1
8/Fc	250 neurons	300 neurons	300 neurons
9/ReLU	250 neurons	300 neurons	300 neurons
10/Fc	6 neurons	6 neurons	6 neurons

(Table 4), leading this advantage to lower memory consumption, computational cost and simpler pipeline.

Table 5
Number of 3D points analyzed in two different scenarios.

Area	Points
Urban	129,553,905
Road	145,759,301

4. Results

In this section, the performance of the traffic sign detection and classification methodologies are presented.

4.1. Acquisition hardware

The LYNX Mobile Mapper by Optech was used for the collection of the data (Puente et al., 2013b). The methodology presented in Sections 3.1 and 3.2 was tested in two different scenarios. The first one is an urban area, that comprises 2.5 km three-lane avenue that encircles the city center of Lugo, in northwest Spain. The second one is a road environment that includes 7.5 km section of AP-9 highway and N-552, N-554 roads in the outskirts of Vigo. The number of 3D points that were analyzed for each scenario, as noted in Soilán et al. (2016) can be found in Table 5.

4.2. Traffic sign detection results

The traffic sign detection process was evaluated using the urban and road areas of the study case. The ground truth was created by manually annotating the position of the traffic signs in these areas. The ground truth is compared with the output of the road sign detection algorithm for traffic signs, which is a set of 3D point clusters, *C*. The evaluation is carried out using Precision, Recall and F1-score for measuring the performance. The results, based in Soilán et al. (2016) are shown in Table 6 together with a comparison with Riveiro et al. (2016) and Wen et al. (2016) results.

4.3. RGB processing results

Finally, regarding the projection of traffic sign points in RGB images, a data reduction metric is provided which shows the quality of the image cropping process and aim to prove that the 3D point cloud processing highly diminishes the non-meaningful data to be analyzed by a 2D TSRS. A ratio that compares the total number of images available over the number of images obtained after the

Table 4
Proposed DNN information compared with previous state-of-the-art methods.

Paper	Data augment. or jittering	# trainable parameters	# ConvNets
Ours	No	14,629,801	1
Jin et al. (2014)	Yes	~ 23 millions	20 (ensemble)
Cireşan et al. (2012)	Yes	~ 90 millions	25 (committee)

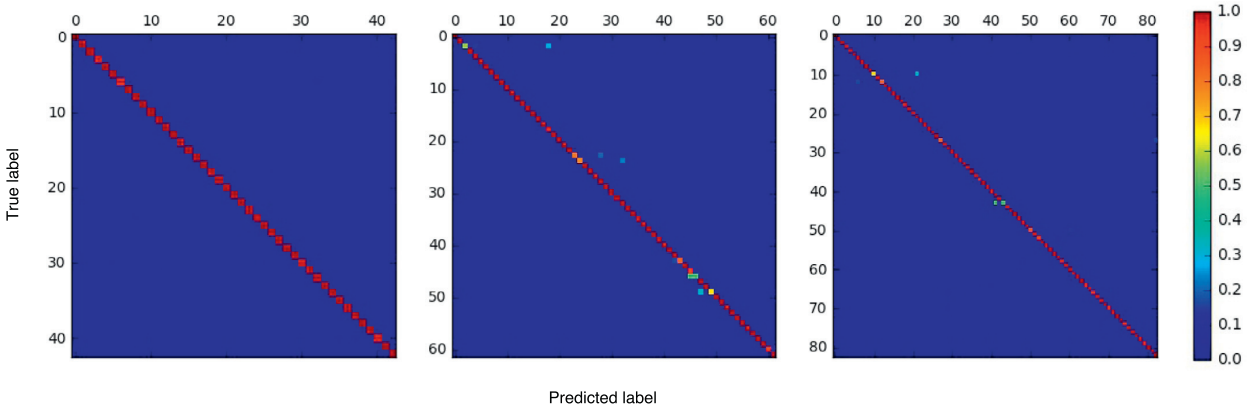


Fig. 8. Confusion matrices. GTSRB on the left, BTSC in the middle and Mixset on the right.

Table 6
Traffic sign detection results.

Area	Precision (%)	Recall (%)	F1 score (%)
Urban	86.1	95.4	90.5
Road	92.8	100	96.3
Global performance			
This paper	89.7	97.9	93.4
Riveiro et al. (2016)	91.3	90.9	91.1
Wen et al. (2016)	91.92	90.53	91.22

Table 7
GTSRB, BTSC and Mixset precision, recall and f1-score recognition results. Mixset includes the cross-validation percentage.

Dataset	Precision (%)	Recall (%)	F1 score (%)
GTSRB	99.71	99.71	99.71
BTSC	98.95	98.87	98.86
Mixset	99.37 ± 0.03	99.36 ± 0.03	99.35 ± 0.03

projection of the 3D points of sign panel was computed, obtaining a value of 5.275.

4.4. Traffic sign recognition results

The following subsections describe the experiments and achieved results in the GTSRB dataset, BTSC dataset and Mixset dataset. As development tools, Torch scientific computer framework[3] and an implementation of spatial transformer networks[4] were used. Overall recognition results of each dataset are shown in Table 7 and confusion matrices in Fig. 8.

4.4.1. GTSRB dataset results

Firstly, to find empirically the best DNN architecture, GTSRB dataset was used in the execution of more than 200 experiments run during 10 epochs with a wide range of DNN configurations composed by the layers described in Section 3.3.3. Each of them consists of 39,209 training images, 12,630 validation traffic signs, a base learning rate fixed to 0.01 and a vanilla Stochastic Gradient Descent algorithm (SGD) as loss function optimizer.

Secondly, top-10 DNN configurations were revised and executed again increasing the number of epochs to 26 expecting to improve accuracy results. Nevertheless, in some cases the accuracy of the DNNs trained grew a little and in other cases it was the same. The best one reached an accuracy of 99.71% in GTSRB, whose configuration is the DNN architecture deeply detailed in Section 3.3.3. It outperforms several GTSRB methods used previously (Table 8). By the time of writing this paper our proposed DNN is top-1 in the GTSRB out of the previously published works.

4.4.2. BTSC dataset results

The Belgian traffic sign classification dataset (Mathias et al., 2013) has 4533 training images and 2562 validation ones split into 62 traffic sign types. Even though an adaptation of this dataset was handcrafted to populate the Mixset showed off in Section 3, in the current subsection experiment the original dataset was used without any further modification in order to measure the performance of the DNN proposed. Considering that this dataset has different traffic sign pictograms, lighting conditions, occlusions, image resolutions and so on than in the GTSRB dataset, our DNN configuration achieves an accuracy of 98.87% (Table 9).

4.4.3. Mixset dataset results

Mixset dataset was generated using the original images from the GTSRB dataset, the adapted ones from the BTSC dataset and the ones from the Spanish dataset. As a result, Mixset consists of 44,130 training traffic sign images and 15,345 validation ones. To evaluate the performance of our DNN in this dataset, five models were trained and tested corresponding each one to a cross-validation fold. The DNN model reaches an average accuracy of 99.36 ± 0.03% being the second fold used in the cross-validation the best one (Table 10). Even though we have a highly imbalanced dataset, the DNN performs well classifying traffic signs that belong to categories with a small number of training instances (Table 11). Some misclassified samples are shown in Fig. 9.

4.5. Processing time

Detection processing times are shown in Table 12. A section of point cloud data of the urban dataset was selected and the methodology presented in Section 3.1 was applied several times to get the average execution time for each algorithm within the processing chain. It was tested using an Intel Core i7-4771 CPU at 3.5 GHz. It can be seen that the ground segmentation process is the most demanding, and the whole processing of almost 30 million points takes about four minutes.

Regarding traffic sign recognition, experiments were performed in a computer built with an Intel Core i7-6700k CPU, 16 GB of RAM and a Nvidia Geforce GTX 1070 discrete GPU which has 1920 CUDA cores and 8 GB of RAM. Training and testing execution times are shown in Table 13.

[3] http://torch.ch/ (accessed 17.03.22).
[4] https://github.com/qassemoquab/stnbhwd (accessed 17.03.22).

Table 8
Recognition rate of different methods on GTSRB dataset.

Paper	Method	Accuracy (%)
Ours	CNN with 3 STNs	99.71
Jin et al. (2014)	HLSGD (20 CNNs ensemble)	99.65
Cireşan et al. (2012)	MCDNN (25 CNNs committee)	99.46
Yu et al. (2016)	GDBM	99.34
Jurisic, Filkovic, and Kalafatic (2015)	OneCNN	99.11 ± 0.10
Stallkamp et al. (2011)	Human performance (avg.)	98.84
Mathias et al. (2013)	INNLP+INNC(I,PI,HOGs)	98.53

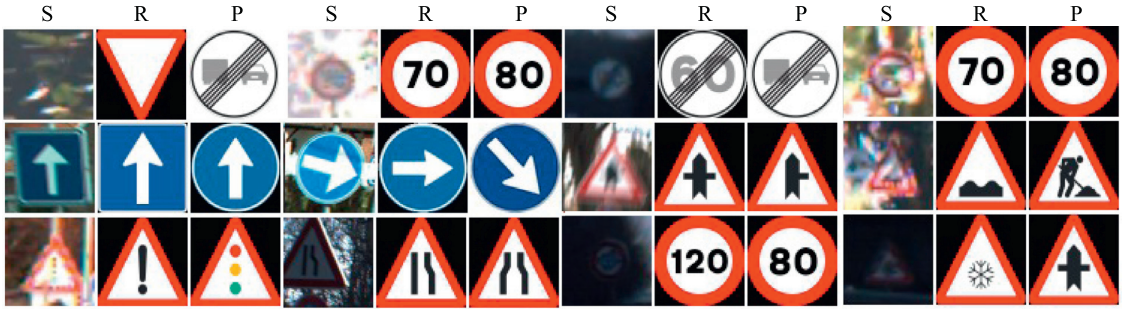


Fig. 9. Misclassified samples. Some misclassified samples of the Mixset model trained. As may be seen, the main reason behind them are occlusions and blurred pictographs, being their recognition even hard for the human visual system. Columns labeled with *S* refer to sample, *R* to real traffic sign category and *P* to prediction.

Table 9
Recognition rate of different methods on BTSC dataset.

Paper	Method	Accuracy (%)
Yu et al. (2016)	GDBM	98.92
Ours	CNN with 3 STNs	98.87
Jurisic et al. (2015)	OneCNN	98.17 ± 0.22
Mathias et al. (2013)	INNLP+SRC(PI)	97.83

Table 10
Mixset model cross-validation results.

Fold	Precision (%)	Recall (%)	F1 score (%)
1	99.37	99.36	99.34
2	99.40	99.38	99.38
3	99.36	99.34	99.34
4	99.33	99.32	99.30
5	99.40	99.38	99.38
Avg.	99.37 ± 0.03	99.36 ± 0.03	99.35 ± 0.03

Table 11
Second fold results of Mixset model for categories with a small size of training examples. The first column represents those categories which contains a determined number of training samples included in the range [*Min–Max*].

[Min–Max]	Avg. precision (%)	Avg. recall (%)	Avg. F1 score (%)
[4–20]	99.47	93.28	95.60
[21–50]	99.14	98.33	98.65
[51–100]	97.48	99.03	98.15
[101–500]	98.97	99.14	99.04
[501–1000]	99.33	99.58	99.45
[1001–1500]	99.65	98.62	99.13
[1501–2000]	98.82	99.92	99.36
[2001–2504]	99.83	99.78	99.81

Table 12
Traffic sign detection processing time.

Algorithm	Time (s)	# Input points
Preprocessing	13.75	28,032,301
Ground Segmentation	117.97	20,440,211
Detection	77.6	17,127,358
Image Projection	25.86	6868
Total	240.34	28,032,301

Table 13
Processing time needed by the DNN proposed to train and test 1 sample.

Process	Time (ms)
Learn 1 sample	11.18 ± 0.02
Test 1 sample	4.28 ± 0.02

5. Conclusions and future work

In this paper a method for the automatic detection and recognition of vertical traffic signs is presented. 3D point clouds collected by a Mobile Mapping System are processed in order to detect traffic sign panels using both geometric and radiometric features. The 3D data are projected on 2D images given the spatio-temporal relationship between the laser scanners and the images taken by the RGB cameras. The images that contain traffic signs are properly cropped and classified using a single DNN that alternates convolutional and spatial transformer modules. Although there are other approaches that combine LiDAR techniques and 2D imagery (Tan, Wang, Wu, Wang, & Pan, 2016; Wen et al., 2016; Yu et al., 2016) our methodology outperforms the previous ones.

The traffic sign detection methodology is tested in different scenarios in Spain, obtaining a F1-score of 93.4%. Projecting the 3D traffic signs detected in the LiDAR point cloud on 2D images drastically reduces the amount of data which is fed to the Traffic Sign Recognition System. For traffic sign classification, we propose and analyze the performance of a single DNN on multiple traffic sign classification datasets. It outperforms previous state-of-the-art methods reporting a recognition rate accuracy of 99.71% in the GTSRB. Also, the DNN avoids the need of handcrafted data augmentation and jittering used in prior approaches (Cireşan et al., 2012; Jin et al., 2014; Sermanet & LeCun, 2011). Moreover, there is less memory requirements and the network has less number of parameters to learn compared with existing methods since we keep away from using several ConvNets in an ensemble or in a committee way.

The main drawback of this method is that it cannot lead to real time applications, as 3D point cloud processing is computationally intensive. Furthermore, setting up the Mobile Mapping System

is expensive and complex. The calibration of the cameras has to be precise, as well as the geometric transformations with respect to the positioning system, where measuring errors of centimeters may lead to large accuracy losses when a 3D point cloud is projected on 2D imagery. Regarding to the traffic sign classification system, the DNN proposed needs a huge amount of traffic sign samples of many categories, taken by cameras with different lighting and weather conditions (fog, rain, sun glare), occlusions, bad viewpoints, faded colors, etc., in order to train a robust model that could cope well with such issues. This is a disadvantage with respect to computer vision approaches based on color and shape feature engineering since such methods do not need any prior knowledge of traffic signs.

The main contributions of this work are four-fold: (1) The methodology presents state-of-the-art results for traffic sign detection through 3D point clouds processing and classification in 2D imagery by means of a DNN, both integrated in the same automated framework. (2) We provide an insight into the proposed DNN capabilities and how do spatial transformer modules work with traffic signs. (3) Multiple public available traffic sign classification datasets are analyzed and used by the classification model, including a dataset with traffic sign images from three European countries. (4) A scalable, publicly available dataset containing around 1500 images of Spanish traffic signs. These contributions lead to practical applications such as automated inventory and maintenance of vertical signage using a data source (i.e. 3D point clouds) which can be simultaneously processed in order to detect a wide range of infrastructure elements, feeding road network information layers to a spatial database. Furthermore, the classification model on its own can be used for real time TSRS since its inference time is quite low and it can be deployed as a standalone service. For instance, expert systems as self-driving cars could benefit from this classification system once the traffic sign has been detected.

Future work should study the impact of different loss function optimizers for ConvNets, other kind of non-linearity layers, dropout layers, and state-of-the-art ConvNets architectures for image recognition like ResNet (He, Zhang, Ren, & Sun, 2016) or Inception (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017) along with spatial transformer networks. Finally, DNN for traffic sign detection should be further investigated in order to build cost-effective car-mounted devices that handle similar pipelines in real time.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness and European Regional Development Fund (ERDF) through the Projects “HERMES – Smart Citizen” (TIN2013-46801-C4-1-R) and “HERMES – S3D” (TIN2013-46801-C4-4-R).

References

Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Network, 32*, 333–338. doi:10.1016/j.neunet.2012.02.023.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., et al. (2011). On the segmentation of 3D LIDAR point clouds. In *Proceedings of 2011 IEEE international conference on robotics and automation*. doi:10.1109/icra.2011.5979818.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise* (pp. 226–231). AAAI Press.

European Union Road Federation (2015). An ERF position paper for maintaining and improving a sustainable and efficient road network. *Technical report*. http://www.erf.be/images/Road-Asset-Management-for_web_site.pdf.

González-Jorge, H., Riveiro, B., Armesto, J., & Arias, P. (2013). Evaluation of road signs using radiometric and geometric data from terrestrial LiDAR. *Optica Applicata, 43*(3), 421–433. doi:10.5277/oa130302.

Gressin, A., Mallet, C., Demantké, J., & David, N. (2013). Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS Journal of Photogrammetry and Remote Sensing, 79*, 240–251. doi:10.1016/j.isprsjprs.2013.02.019.

Gudigar, A., Chokkadi, S., Raghavendra, U., & Acharya, U. R. (2017). Local texture patterns for traffic sign recognition using higher order spectra. *Pattern Recognition Letters, 94*, 202–210. doi:10.1016/j.patrec.2017.02.016.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of 2016 IEEE conference on computer vision and pattern recognition (CVPR)*. doi:10.1109/cvpr.2016.90.

Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In *Proceedings of advances in neural information processing systems* (pp. 2017–2025).

Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition?. In *Proceedings of 2009 IEEE 12th international conference on computer vision*. doi:10.1109/iccv.2009.5459469.

Jin, J., Fu, K., & Zhang, C. (2014). Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems, 15*(5), 1991–2000. doi:10.1109/tits.2014.2308281.

Jurisic, F., Filkovic, I., & Kalafatic, Z. (2015). Multiple-dataset traffic sign classification with OneCNN. In *Proceedings of 2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. doi:10.1109/acpr.2015.7486576.

Kaplan Berkaya, S., Gunduz, H., Ozsen, O., Akinlar, C., & Gunal, S. (2016). On circular traffic sign detection and recognition. *Expert Systems with Applications, 48*, 67–75. doi:10.1016/j.eswa.2015.11.018.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE, 86*, 2278–2324. doi:10.1109/5.726791.

Maldonado-Bascón, S., Acevedo-Rodríguez, J., Lafuente-Arroyo, S., Fernández-Caballero, A., & López-Ferreras, F. (2010). An optimization on pictogram identification for the road-sign recognition task using SVMs. *Computer Vision and Image Understanding, 114*(3), 373–383. doi:10.1016/j.cviu.2009.12.002.

Mathias, M., Timofte, R., Benenson, R., & Van Gool, L. (2013). Traffic sign recognition How far are we from the solution?. In *Proceedings of the 2013 international joint conference on neural networks (IJCNN)*. doi:10.1109/ijcnn.2013.6707049.

Oliveira, L., Nunes, U., Peixoto, P., Silva, M., & Moita, F. (2010). Semantic fusion of laser and vision in pedestrian detection. *Pattern Recognition, 43*(10), 3648–3659. doi:10.1016/j.patcog.2010.05.014.

Pu, S., Rutzinger, M., Vosselman, G., & Elberink, S. O. (2011). Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing, 66*(6), S28–S39. doi:10.1016/j.isprsjprs.2011.08.006.

Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013a). Review of mobile mapping and surveying technologies. *Measurement, 46*(7), 2127–2145. doi:10.1016/j.measurement.2013.03.006.

Puente, I., González-Jorge, H., Riveiro, B., & Arias, P. (2013b). Accuracy verification of the Lynx Mobile Mapper system. *Optics & Laser Technology, 45*, 578–586. doi:10.1016/j.optlastec.2012.05.029.

Riveiro, B., Díaz-Vilarino, L., Conde-Carnero, B., Soilán, M., & Arias, P. (2016). Automatic segmentation and shape-based classification of retro-reflective traffic signs from mobile LiDAR data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9*(1), 295–303. doi:10.1109/jstars.2015.2461680.

Sermanet, P., & LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *Proceedings of the 2011 international joint conference on neural networks*. doi:10.1109/ijcnn.2011.6033589.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., & Arias, P. (2016). Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *ISPRS Journal of Photogrammetry and Remote Sensing, 114*, 92–101. doi:10.1016/j.isprsjprs.2016.01.019.

Spanish Government (2003). BOE.es - Documento BOE-A-2000-1546. http://www.boe.es/diario_boe/txt.php?id=BOE-A-2003-23514.

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The German traffic sign recognition benchmark: A multi-class classification competition. In *Proceedings of the 2011 international joint conference on neural networks*. doi:10.1109/ijcnn.2011.6033395.

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Network, 32*, 323–332. doi:10.1016/j.neunet.2012.02.016.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence*, 4278–4284.

Tan, M., Wang, B., Wu, Z., Wang, J., & Pan, G. (2016). Weakly supervised metric learning for traffic sign recognition in a LIDAR-Equipped vehicle. *IEEE Transactions on Intelligent Transportation Systems, 17*(5), 1415–1427. doi:10.1109/tits.2015.2506182.

Timofte, R., Zimmermann, K., & Van Gool, L. (2011). Multi-view traffic sign detection, recognition, and 3D localisation. *Machine Vision and Applications, 25*(3), 633–647. doi:10.1007/s00138-011-0391-3.

Wen, C., Li, J., Luo, H., Yu, Y., Cai, Z., Wang, H., et al. (2016). Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems, 17*(1), 27–37. doi:10.1109/tits.2015.2418214.

Yu, Y., Li, J., Wen, C., Guan, H., Luo, H., & Wang, C. (2016). Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing, 113*, 106–123. doi:10.1016/j.isprsjprs.2016.01.005.

Zaklouta, F., Stanciulescu, B., & Hamdoun, O. (2011). Traffic sign classification using K-d trees and Random Forests. In *Proceedings of the 2011 international joint conference on neural networks*. doi:10.1109/ijcnn.2011.6033494.

CAPÍTULO 3

DEEP NEURAL NETWORK FOR TRAFFIC SIGN RECOGNITION SYSTEMS: AN ANALYSIS OF SPATIAL TRANSFORMERS AND STOCHASTIC OPTIMISATION METHODS

Resumen

Este artículo de investigación presenta un nuevo enfoque para sistemas de reconocimiento de señales de tráfico basado en aprendizaje profundo. Se llevan a cabo varios experimentos de clasificación sobre conjuntos de datos de señales de tráfico de Alemania y Bélgica que están públicamente disponibles, utilizando una red neuronal profunda que contiene capas convolucionales y redes de transformadores espaciales. Dichos ensayos están diseñados para medir el impacto de diversos factores con el objetivo final de diseñar una red neuronal convolucional que pueda mejorar los sistemas de clasificación de señales de tráfico propuestos hasta el momento. En primer lugar, se evalúan diferentes algoritmos de optimización de gradientes descendentes estocásticos adaptativos y no adaptativos, tales como SGD, SGD-Nesterov, RMSprop y

Adam. Posteriormente, se analizan múltiples combinaciones de redes de transformadores espaciales ubicadas en distintas posiciones dentro de la red neuronal principal. La tasa de reconocimiento de la red neuronal convolucional propuesta alcanza una precisión del 99,71 % en el German Traffic Sign Recognition Benchmark (GTSRB), superando los métodos propuestos anteriormente en la literatura, al mismo tiempo que es más eficiente en términos de requisitos de memoria.

Neural Networks 99 (2018) 158–165

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods

Álvaro Arcos-García *, Juan A. Álvarez-García, Luis M. Soria-Morillo

Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, 41012, Sevilla, Spain

HIGHLIGHTS

- A Deep Neural Network that is top-1 ranked in the German traffic sign benchmark.
- Effectiveness analysis of Spatial Transformer Networks for traffic sign recognition.
- Quantitative comparison of several stochastic gradient descent optimisation methods.

ARTICLE INFO

Article history:
Received 25 July 2017
Received in revised form 15 November 2017
Accepted 18 January 2018
Available online 31 January 2018

Keywords:
Deep learning
Traffic sign
Spatial transformer network
Convolutional neural network

ABSTRACT

This paper presents a Deep Learning approach for traffic sign recognition systems. Several classification experiments are conducted over publicly available traffic sign datasets from Germany and Belgium using a Deep Neural Network which comprises Convolutional layers and Spatial Transformer Networks. Such trials are built to measure the impact of diverse factors with the end goal of designing a Convolutional Neural Network that can improve the state-of-the-art of traffic sign classification task. First, different adaptive and non-adaptive stochastic gradient descent optimisation algorithms such as SGD, SGD-Nesterov, RMSprop and Adam are evaluated. Subsequently, multiple combinations of Spatial Transformer Networks placed at distinct positions within the main neural network are analysed. The recognition rate of the proposed Convolutional Neural Network reports an accuracy of 99.71% in the German Traffic Sign Recognition Benchmark, outperforming previous state-of-the-art methods and also being more efficient in terms of memory requirements.

1. Introduction

Traffic sign recognition systems (TSRS) are essential in many real-world applications such as autonomous driving, traffic surveillance, driver safety and assistance, road network maintenance, and analysis of traffic scenes. Normally, a TSRS concerns two related subjects which are traffic sign detection (TSD) and traffic sign recognition (TSR). The former focuses on the localisation of the targets in the pictures while the latter performs a fine-grained classification to identify the type of targets detected (De La Escalera, Moreno, Salichs, & Armingol, 1997).

Traffic signs constitute a fundamental asset within the road network because their aim is to be easily noticeable by pedestrians and drivers in order to warn and guide them during both the day and night. The fact that signs are designed to be unique and to have distinguishable features such as simple shapes and uniform colours implies that their detection and recognition is a constrained problem. Nevertheless, the development of a robust real-time TSRS still presents a challenging task due to real-world variability, such as scale variations, bad viewpoints, motion-blur, faded colours, occlusions, and lightning conditions. On top of that, there are more than 300 different traffic sign categories defined by the Vienna Convention on Road Traffic (United Nations Economic Commission for Europe, 1968). This treaty has been signed by 63 countries, although a few minor visual variations of traffic sign pictographs still exist between countries, which can lead to complications in the automated recognition task. Any TSRS must cope well with such issues.

The main contributions of this work are four-fold: (1) A state-of-the-art traffic sign recognition system based on a Convolutional Neural Network (CNN) that includes Spatial Transformer Networks (STN) and outperforms previously published work related with the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp, Schlipsing, Salmen, & Igel, 2011); (2) An insight into the proposed CNN capabilities along with the performance impact of spatial transformer layers within the network; (3) Analysis of the effect

* Corresponding author.
E-mail addresses: aarcos1@us.es (Á. Arcos-García), jaalvarez@us.es (J.A. Álvarez-García), lsoria@us.es (L.M. Soria-Morillo).

https://doi.org/10.1016/j.neunet.2018.01.005
0893-6080/

of diverse gradient descent optimisation algorithms on the CNN presented. (4) Multiple publicly available European traffic sign classification datasets are reviewed and evaluated by the CNN. These contributions lead to practical applications, such as self-driving cars and automated inventory and maintenance of vertical signage, since the CNN can perform fine-grained classification once the traffic sign has been detected. Moreover, as the CNN outperforms the human visual system, its inference time is low and can also be deployed as a stand-alone service, it can therefore be used in real-time applications.

The rest of the paper is organised as follows. Section 2 reviews related works of traffic sign recognition systems. Section 3 describes the experiments conducted to analyse the impact of both spatial transformers and stochastic optimisation algorithms on the proposed CNN. Recognition results are then shown in Section 4. Finally, conclusions are drawn and further work is proposed in Section 5.

2. Related work

Chronologically, approaches of published studies on traffic sign recognition systems have evolved from colour and shape-based methods to machine-learning-based methods. In recent times, Deep Neural Networks (DNN) have attracted attention in pattern recognition and computer vision research, and have been widely adopted for both object detection (Liu et al., 2016; Redmon & Farhadi, 2016; Ren, He, Girshick, & Sun, 2015) and recognition (Huang, Liu, Weinberger, & van der Maaten, 2016; Szegedy, Ioffe, Vanhoucke, & Alemi, 2017), thanks to the release of several publicly available datasets composed of millions of images (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010; Krizhevsky, Sutskever, & Hinton, 2012; Lin et al., 2014). Moreover, DNNs have been applied in autonomous driving related challenges such as car (Huval et al., 2015), lane (Li, Mei, Prokhorov, & Tao, 2017), and pedestrian (Tian, Luo, Wang, & Tang, 2015) detection.

With regard to the traffic sign detection and classification problem domain, colour-based approaches are very common. These methods use different colour spaces for segmentation of the road image, such as RGB (Escalera, Moreno, Salichs, & Armingol, 1997), HIS (Maldonado-Bascon, Lafuente-Arroyo, Gil-Jimenez, Gomez-Moreno, & Lopez-Ferreras, 2007), and HSV (Shadeed, Abu-Al-Nadi, & Mismar, 2003). The shape-based method is another popular approach for traffic sign recognition and detection. Symmetry information of circular, triangular, square and octagonal shapes are used in Loy and Barnes (2004), a radial symmetry detector is proposed in Barnes, Zelinsky, and Fletcher (2008), Hough transforms are investigated in Barnes, Loy, and Shaw (2010) and a circular traffic sign recognition system is studied in Kaplan Berkaya, Gunduz, Ozsen, Akinlar, and Gunal (2016). Hence, neither colour nor shape-based techniques, need any prior knowledge of traffic signs and heavily depend on custom-designed algorithms and feature engineering.

One of the main problems before the year 2011 was the lack of publicly available traffic sign datasets. The Belgian Traffic Sign Dataset (BTSD) (Timofte, Zimmermann, & Van Gool, 2011), the German Traffic Sign Recognition and Detection Benchmark (GTSRB and GTSDB) (Stallkamp et al., 2011), the Croatian traffic sign dataset (rMASTIF) (Jurisic, Filkovic, & Kalafatic, 2015), the Dataset of Italian Traffic Signs (DITS) (Youssef, Albani, Nardi, & Bloisi, 2016) and the Tsinghua-Tencent 100 K benchmark (Zhu et al., 2016) solved this issue and boosted research into TSRS since several of these datasets are commonly used to evaluate the performance of computer vision algorithms for traffic sign detection and recognition. These kinds of datasets are crucial to generate robust machine learning and deep learning models as they contain a huge amount of traffic sign samples of multiple categories, taken by cameras with various weather and lighting conditions, occlusions, bad viewpoints, etc.

More recently, machine learning has started to play a key role in the traffic sign classification task. Mathias, Timofte, Benenson, and Van Gool (2013) propose fine-grained classification by applying different methods through a pipeline of three stages: feature extraction, dimensionality reduction and classification. On GTSRB, they reach 98.53% accuracy by merging grey-scale values of traffic sign images and features based on the Histogram of Oriented Gradients (HOG), reducing the dimensionality through Iterative Nearest Neighbours-based Linear Projections (INNLP) and finally classifying with Iterative Nearest Neighbours (INNC) (Timofte & Van Gool, 2015). Although other machine learning algorithms such as Support Vector Machines (SVM) (Salti, Petrelli, Tombari, Fioraio, & Di Stefano, 2015), Random Forests (Zaklouta, Stanciulescu, & Hamdoun, 2011) and Nearest Neighbours (Gudigar, Chokkadi, Raghavendra, & Acharya, 2017) have been widely used to recognise traffic sign images, Convolutional Neural Networks (Lecun, Bottou, Bengio, & Haffner, 1998), also known as ConvNets or CNNs, showed particularly higher classification accuracies in the competition. Neural networks are data driven self-adaptive methods because they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model (Huang, 1996). In addition, there are universal functional approximators in the neural networks that can approximate any function with arbitrary accuracy (Huang, 1999; Huang & Du, 2008). Cireşan, Meier, Masci, and Schmidhuber (2012) won the GTSRB contest (Stallkamp, Schlipsing, Salmen, & Igel, 2012) with 99.46% accuracy thanks to a committee of 25 CNNs by using data augmentation and jittering. Sermanet and LeCun (2011) used a multi-scale CNN and achieved an accuracy of 98.31%, thereby granting them second place in the GTSRB challenge. Later, Jin, Fu, and Zhang (2014) proposed a hinge loss stochastic gradient descent method to train an ensemble of 20 CNNs that resulted in 99.65% accuracy and offered a faster and more stable convergence than previous work. However, these approaches can still be improved through the avoidance of the use of hand-crafted data augmentation and of the application of multiple CNNs in an ensemble or via a committee for the reason that these normally lead to higher memory and computation costs.

3. Methodology

In this work, we propose a traffic sign recognition system that carries out fine-grained classification of traffic sign images through a CNN whose main blocks are convolutional and spatial transformer modules. In order to find an accurate and efficient CNN for such a purpose, the effect of using several STNs and different stochastic gradient descent optimisation methods are researched and discussed.

3.1. Dataset and data pre-processing

Several publicly available traffic sign datasets have been gathered in countries such as the United States (Mogelmose, Trivedi, & Moeslund, 2012), Belgium (Timofte et al., 2011), Germany (Stallkamp et al., 2011), Croatia (Jurisic et al., 2015), Italy (Youssef et al., 2016), Sweden (Larsson & Felsberg, 2011), and China (Zhu et al., 2016).

This paper focuses on both the spatial transformer effectiveness and cost function optimisation experiments on the GTSRB (Stallkamp et al., 2011) dataset. There are multiple reasons for choosing this dataset over the others, including the fact that it is highly accepted and is used for comparing traffic sign recognition approaches in the literature. Moreover, its authors and the organisation behind them held a public competition whereby scientists from different fields contributed with their results and tested the GSTRB dataset. Nowadays, a GTSRB website is maintained where

Fig. 1. GTSRB dataset pre-processed.

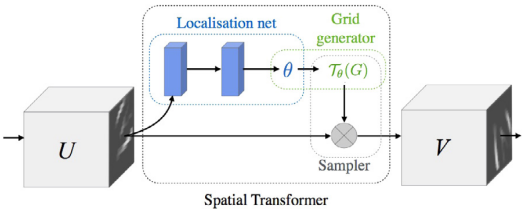


Fig. 2. Spatial transformer network components (Jaderberg et al., 2015).

submissions of results are still accepted, processed and shown in a leaderboard. Such ranking helps to find out which are the state-of-the-art methodologies utilised for the task of traffic sign classification. Last but not least, the GTSRB dataset contains traffic sign samples with different resolutions and image distortions that were extracted from 1-second video sequences. These samples each belong to one of the 43 existing classes. Its ground truth data is reliable due to its semi-automatic annotation, the training set has 39,209 images, and the validation set consists of 12,630 images, which are used to measure the performance of the algorithms. Traffic sign samples are raw RGB images whose size varies from 15×15 to 250×250 pixels.

During the pre-processing stage, all the samples are down-sampled or up-sampled to 48×48 pixels, and both global normalisation and local contrast normalisation with Gaussians kernels (Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009) are computed for the purpose of centring each input image around its mean value as well as for the enhancement of the edges (Fig. 1).

3.2. Convolutional neural network architecture

Inspired by the approach by Cireşan et al. (2012), the proposed method for the recognition of traffic signs is a single CNN that combines several types of layers: convolutional, spatial transformer (Jaderberg, Simonyan, Zisserman, et al., 2015), Rectified Linear Units (ReLU) (Nair & Hinton, 2010), local contrast normalisation (Jarrett et al., 2009) and max-pooling (Scherer, Müller, & Behnke, 2010). These layers act as a feature extractor that maps raw pixel information of the input image to a tensor which is classified later into a particular traffic sign category by two fully connected layers. All variable parameters of these layers are optimised together through the minimisation of the misclassification error over the GTSRB training set.

The convolutional layers carry out a 2-dimensional convolution of their $n-1$ input maps with a filter of size $F_x^n \times F_y^n$, where x and y represent the size of each dimension. Each convolutional layer is composed of neurons which have learnable biases and weights. During the feed-forward process of the neural network, each filter is convolved across the height and width of the input map, and a dot product is performed that produces a 2-dimensional activation map of that filter. The resulting activations of the n output maps are given by the sum of the $n-1$ convolutional responses, which are passed through a non-linear activation function f, that is computed by a ReLU layer in our case, where n is the convolutional layer, i and j represent the input map and the output map respectively, a indicates a map of size $x \times y$, the weights w_{ij} are represented as a filter of size $F_x \times F_y$ which connects the input map with the output map, and b_j is the bias of the output map (Eq. (1)).

$$a_j^n = \sum_{i=1}^{n-1} a_i^{n-1} * w_{ij}^n + b_j^n. \tag{1}$$

ReLU layers (Nair & Hinton, 2010) are made up of neurons that apply the activation function $f(x) = max(0, x)$, where x is the input to a neuron. These layers enhance the non-linear properties of the network, including the decision function, without affecting the learnable parameters of the convolutional layer.

Local contrast normalisation layers (Jarrett et al., 2009) normalise the contrast of an input map through subtractive local normalisation and divisive local normalisation. Both operations use a Gaussian kernel, and are computed in local spatial regions of the input map on a per-feature basis.

Max-pooling layers (Scherer et al., 2010) progressively reduce the spatial size of the feature maps, by directly decreasing the amount of parameters along with computation costs. Moreover, these layers control overfitting by selecting superior invariant features and generalisation is improved. The output of this layer is given by the maximum activation over non-overlapping regions of filter size $F_x \times F_y$, where the input map is downsampled by a factor of F_x and F_y along both width and height, although depth dimension remains unchanged.

Fully connected layer neurons have full connections to all activations in the previous layer and therefore they combine the outputs of the previous layer into a 1-dimensional feature vector. The last fully-connected layer of the network performs the classification task since it has one output neuron per class, followed by a logarithmic softmax activation function.

Spatial transformer units (Jaderberg et al., 2015) aim to perform a geometric transformation on an input map so that CNNs are provided with the ability to be spatially invariant to the input data in a computationally efficient manner. Thanks to such transformations, there is no need for extra training supervision, hand-crafted data augmentation (such as rotation, translation, scaling, skewing, cropping), or dataset normalisation techniques. This differentiable module can be inserted into existing CNN architectures since the parameters of the transformation that are applied to feature maps are learnt by means of a back-propagation algorithm. Spatial transformer networks consist of 3 elements: the localisation network, the grid generator and the sampler (Fig. 2).

The localisation network $f_{loc}()$ takes an input feature map $U \in R^{H \times W \times C}$, where H, W and C are the height, width and channels respectively, and outputs the parameters θ of the transformation T_θ to be applied to the feature map $\theta = f_{loc}(U)$. The dimension of θ depends on the transformation type T_θ that is being parameterised: this is 6-dimensional in our proposed network since it performs a 2D affine transformation A_θ, which allows translation, cropping, rotation, scaling, and skewing. The localisation network can comprise any number of convolutional and fully connected layers and must have at least one final regression layer with 6 output neurons in order to generate the transformation parameters θ. It should be borne in mind that this final output layer is initialised with the identity transformation matrix. Such parameters are used by the grid generator to create a sampling grid, which is a set of points where the input map has to be sampled to obtain the desired transformed output. Finally, the sampler uses the sampling grid and the input feature map U as inputs in order to perform a bilinear sampling, which produces the transformed output feature map

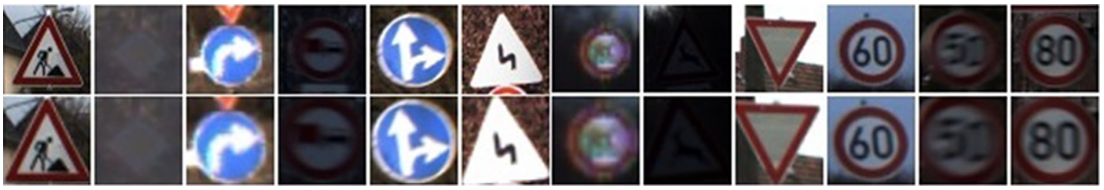


Fig. 3. Spatial transformer network. Input images above and output images below after computing affine transformations.

$V \in R^{H' \times W' \times C}$, where H', W' are the height and width of the sampling grid respectively.

For source coordinates in the input feature map (x_i^s, y_i^s) and a learnt 2D affine transformation matrix A_θ, the target coordinates of the regular grid in the output feature map (x_i^t, y_i^t) are given as follows (Eq. (2)):

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}. \tag{2}$$

As regards traffic sign recognition, spatial transformer networks learn to focus on the traffic sign by gradually removing geometric noise and background so that only the interesting zones of the input are forwarded to the next layers of the network (Fig. 3). To the best of our knowledge, no peer review work has been published that has included the spatial transformer unit in a CNN for the traffic sign recognition task.

In order to measure the performance of spatial transformer layers for traffic sign recognition, we set the main CNN architecture shown in Table 1, which contains no STN. This CNN progressively increases the number of feature maps extracted from the input image through convolutional layers. At the same time, the input image's dimension is reduced by max-pooling layers and therefore the network is able to extract features on different scales. Finally, a fully-connected layer performs the classification of the traffic sign fed into the network. The stride of each convolutional layers is set to 1 in order to leave all spatial down-sampling computation to max-pooling layers, and zero-padding is set to 2. Regarding max-pooling layers, their stride is set to 2 and zero-padding to 0. Input and output feature maps of convolutional layers, as well as kernel sizes, are fixed.[1]

Due to the possibility of combining up to 3 STNs in different parts of the CNN, several network architectures were set in order to measure their influence in the final result. Note that no more than three STNs are included in the analysis since the size of output feature maps of the subsequent network's layers could not be further decreased. Progressively, spatial transformer modules are added immediately before the convolutional layers of the main network. The localisation network of the three spatial transformer layers is built with a max-pooling layer followed by two blocks of convolutional, ReLU and max-pooling, and finally, two fully-connected layers joined by a ReLU unit. The output of the last fully-connected layer consists of 6 neurons, which correspond to the parameters of the affine transformation matrix. Detailed architectures of localisation networks are drawn in Table 2. Analogous to the configuration of convolutional layers, kernel sizes and the number of input and output feature maps are fixed.

In total, there are eight different CNN architectures as a result of the possible combinations described. To denote such configurations, on one hand, c refers to a convolutional block which includes convolutional, ReLU, max-pooling and local contrast normalisation layers. On the other hand, s_i indicates the ith configuration of a spatial transformer module. For instance, a network with only one spatial transformer at the beginning is expressed as $s_1\_c\_c\_c$. Note that s_1 can only be placed before the first convolutional layer, s_2 ahead of the second convolutional unit, and s_3 preceding the third convolutional module.

[1] https://github.com/aarcosg/tsr-torch.

Table 1
Main CNN architecture without spatial transformer modules.

Layer	Type	# Maps & neurons	Kernel
0	Input	3 m. of 48 × 48 n.	
1	Convolutional	200 m. of 46 × 46 n.	7 × 7
2	ReLU	200 m. of 46 × 46 n.	
3	Max-Pooling	200 m. of 23 × 23 n.	2 × 2
4	Local Contrast Norm.	200 m. of 23 × 23 n.	
5	Convolutional	250 m. of 24 × 24 n.	4 × 4
6	ReLU	250 m. of 24 × 24 n.	
7	Max-Pooling	250 m. of 12 × 12 n.	2 × 2
8	Local Contrast Norm.	250 m. of 12 × 12 n.	
9	Convolutional	350 m. of 13 × 13 n.	4 × 4
10	ReLU	350 m. of 13 × 13 n.	
11	Max-Pooling	350 m. of 6 × 6 n.	2 × 2
12	Local Contrast Norm.	350 m. of 6 × 6 n.	
13	Fully connected	400 neurons	1 × 1
14	ReLU	400 neurons	
15	Fully connected	43 neurons	1 × 1
16	Softmax	43 neurons	

Table 2
Localisation network details of spatial transformers used in the basic CNN. Kernel size of convolutional layers is set to 5 × 5 and max-pooling layers to 2 × 2. The annotation shown in the table is simplified, for instance, 3 of 48 × 48 stand for 3 feature maps of 48 × 48 neurons each.

Layer/Type	Loc. net of ST 1	Loc. net of ST 2	Loc. net of ST 3
0/Input	3 of 48 × 48	200 of 23 × 23	250 of 12 × 12
1/Max-Pool	3 of 24 × 24	200 of 11 × 11	250 of 6 × 6
2/Conv	250 of 24 × 24	150 of 11 × 11	150 of 6 × 6
3/ReLU	250 of 24 × 24	150 of 11 × 11	150 of 6 × 6
4/Max-Pool	250 of 12 × 12	150 of 5 × 5	150 of 3 × 3
5/Conv	250 of 12 × 12	200 of 5 × 5	200 of 3 × 3
6/ReLU	250 of 12 × 12	200 of 5 × 5	200 of 3 × 3
7/Max-Pool	250 of 6 × 6	200 of 2 × 2	200 of 1 × 1
8/Fc	250 neurons	300 neurons	300 neurons
9/ReLU	250 neurons	300 neurons	300 neurons
10/Fc	6 neurons	6 neurons	6 neurons

3.3. Stochastic gradient descent optimisation algorithms

Optimisation is the process of finding the set of parameters w that minimise the loss function L. The loss function L quantifies the quality of a particular set of parameters w based on how well the inferred scores match the ground truth labels in the training data. In this work, the last layer of the CNNs proposed is a softmax classifier that uses the cross-entropy loss function (Eq. (3)), where i enumerates the different classes, y is the predicted probability distribution, and y' is the true distribution represented as a one-hot vector. The softmax function (Eq. (4)) is employed to compute y. It takes a K-dimensional vector of arbitrary real-valued scores z and squashes it to a K-dimensional vector $f(z)$ of values in the range $(0, 1]$ that add up to 1, where j represents the jth element of

Table 3
Configuration parameters of stochastic gradient descent optimisation algorithms.

SGD w/o momentum	SGD with Nesterov
Momentum = 0 Weight decay = 0 Learning rate = 1e−2	Momentum = 0.9 Weight decay = 1e−4 Nesterov Learning rate = 1e−3
RMSprop	**Adam**
$\alpha = 0.99$ $\epsilon = 1e{-}8$ Weight decay = 0 Learning rate = 1e−5	$\beta 1 = 0.9$ $\beta 2 = 0.999$ $\epsilon = 1e{-}8$ Weight decay = 0 Learning rate = 1e−4

the vector f.

$$H_{y'}(y) = -\sum_i y'_i \log(y_i)$$
$$y_i \in (0, 1) : \sum_i y_i = 1 \forall i \tag{3}$$

$$f_j(z) = \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_k}}. \tag{4}$$

Gradient descent is the most common and established algorithm for the optimisation of the neural network's loss function. Iteratively, it computes the gradient of the objective function L with respect to the model's parameters w and then updates them. One of its variants is the mini-batch gradient descent that can be written as follows:

$$w_{k+1} = w_k - \eta_k \check{\nabla} L(w_k). \tag{5}$$

This computes the gradient $\check{\nabla} L(w_k) := \nabla L(w_k; x_k^{(i:i+n)}; y_k^{(i:i+n)})$ of the loss function L and performs an update for every mini-batch of n training examples $x^{(i)}$ and labels $y^{(i)}$, where η represents the learning rate.

In order to accelerate training, certain techniques, such as Nesterov's Accelerated Gradient method (NAG) (Nesterov, 1983), and Polyak's heavy-ball method (HB) (Polyak, 1964), have been widely used. These can be categorised as stochastic momentum methods.

Adaptive optimisation methods constitute another family of gradient descent algorithms. In contrast to non-adaptive methods, they perform local optimisation by choosing a local distance measure constructed from the history of iterates $w_1, \ldots, w_k$. Examples in this category include the Adaptive Gradient algorithm (AdaGrad) (Duchi, Hazan, & Singer, 2011), Root Mean Square Propagation (RMSprop) (Tieleman & Hinton, 2012), and Adaptive Moment Estimation (Adam) (Kingma & Ba, 2015).

In this paper, we compare the effectiveness of four mini-batch gradient descent optimisation algorithms applied to the CNNs proposed in Section 3.2: Stochastic Gradient Descent (SGD) without momentum (Qian, 1999), SGD with Nesterov's accelerated gradient, RMSprop, and Adam.

For hyper-parameter tuning, several networks were trained for several epochs in order to find an adequate initial learning rate value that reaches model convergence. We observed that a high learning rate such as 0.01 fails to work well in the cases of RMSprop and Adam, since it achieves low accuracy scores. The main reason could be that, unlike SGD where the learning rate is fixed and it can optionally follow an annealing schedule, RMSprop and Adam calculate adaptive learning rates for each model's parameter based on the history of iterates. Consequently, a lower learning rate is set for such methods in order to prevent loss values becoming stuck at bad spots in the optimisation landscape. The initial parameters of these algorithms are shown in Table 3.

Table 4
Recognition rate accuracy achieved by CNNs configurations described in Section 3.2 using different loss function optimisers: SGD without momentum (SGD), SGD with Nesterov accelerated gradient (SGD-N), Root Mean Square Propagation (RMSprop) and Adaptive Moment Estimation (Adam). c refers to convolutional block and s to spatial transformer module. Experiments were run for 15 epochs.

CNN/Optimiser	SGD	SGD-N	RMSprop	Adam	# Parameters
c_c_c	98.31	98.33	98.66	98.81	7,303,883
s_1_c_c_c	99.09	99.15	99.37	99.20	11,137,389
c_s_2_c_c	99.22	99.13	99.28	99.15	9,046,339
c_c_s_3_c	99.02	99.04	99.11	99.39	9,053,839
s_1_c_s_2_c_c	99.31	99.30	99.38	99.23	12,879,845
s_1_c_c_s_3_c	99.21	99.25	99.32	99.32	12,887,345
c_s_2_c_s_3_c	99.34	99.23	99.45	99.28	10,796,295
s_1_c_s_2_c_s_3_c	**99.49**	99.43	99.40	99.42	14,629,801

4. Results

Having described the CNN architectures and the loss function optimisers, 32 experiments were run on a computer built with an Intel Core i7-6700k CPU, 16 GB of RAM, and a Nvidia Geforce GTX 1070 discrete GPU which has 1920 CUDA cores and 8 GB of RAM, whereby the Torch scientific computer framework (Collobert, Kavukcuoglu, & Farabet, 2011) and an implementation of spatial transformer networks for Torch (Oquab, 2017) were applied as development tools. The objective is to identify the best places to add the STNs within the CNN at the same time as adding the best stochastic gradient descent optimiser. With a mini-batch size of 50, each experiment is a two-stage process that trains the neural network with the GTSRB training set and then tests it with the GTSRB validation set for 15 epochs. The results presented in Table 4 show the maximum accuracy percentage achieved by each CNN model over the validation set. The best configuration found contains three spatial transformer modules (s_1_c_s_2_c_s_3_c) and the computed loss value is optimised by means of SGD without momentum algorithm. On the other hand, the worst results are obtained by the CNN that includes no spatial transformer (c_c_c) regardless of the optimiser, and the second-worse results are given by the CNN with a spatial transformer located immediately before the last convolutional layer (c_c_s_3_c). It should be borne in mind that the winning configuration contains double the number of the model parameters of the worst CNN. As a consequence, for the SGD without momentum algorithm, the training time per epoch of the CNN with three spatial transformers is 355.05 ± 0.8 s while the CNN with no spatial transformer takes 212.12 ± 0.1 s. To sum up, the inclusion of spatial transformer units into the main CNN leads to superior classification performance, especially when they are added between at least the first layers. This improvement in performance is due to the fact that the spatial transformer scale-normalises and crops out the appropriate traffic sign region, thereby simplifying the subsequent classification task.

By considering such results and choosing the CNN s_1_c_s_2_c_s_3_c (Fig. 4), certain insights related with the comparison of the optimisation algorithms were revealed. Firstly, the solutions obtained by adaptive methods (RMSprop, Adam) generalise worse than those attained by non-adaptive methods (SGD, SGD-N). Early on in training, all four methods achieve nearly perfect training accuracy; however, during testing time, non-adaptive methods outperform adaptive methods in terms of accuracy and they display a more stable behaviour as shown in Fig. 5(b). Secondly, the adaptive methods achieve similar training loss values and lower testing loss values than non-adaptive methods. Nevertheless, their testing performance is worse, which again leads to the idea that non-adaptive algorithms generalise better than adaptive algorithms. Finally, it should be emphasised that Adam and RMSprop required the initial learning rate to be tuned, as detailed in previous section, since

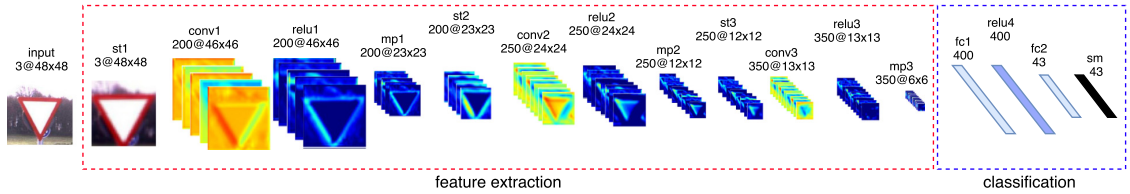
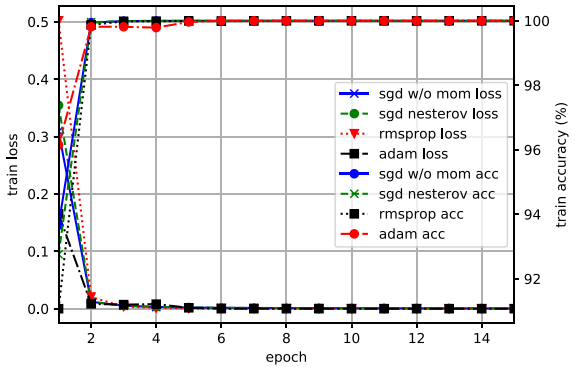
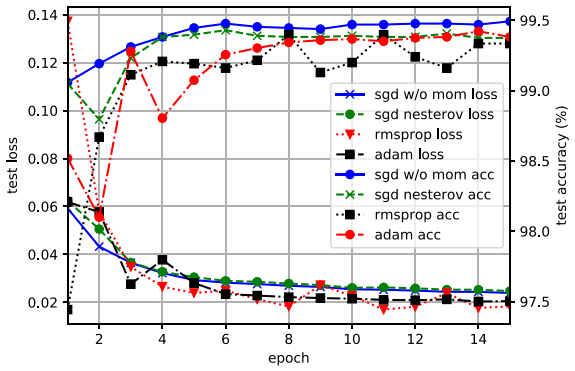


Fig. 4. CNN for traffic sign recognition. Local contrast normalisation layers and the localisation network of spatial transformers have been omitted in the figure above to simplify its visualisation. The *st* layers refer to spatial transformer networks, *conv* to convolutional layers, *mp* to max-pooling layers, *fc* to fully-connected layers, and *sm* to the softmax layer.



(a) GTSRB (Train).



(b) GTSRB (Test).

Fig. 5. Comparison of training loss and testing loss versus accuracy for four different loss function optimisers on applying the CNN model with 3 STNs $s_1\_c\_s_2\_c\_s_3\_c$.

with default settings, they achieved very low accuracy scores in comparison with those of non-adaptive methods. Although these insights should be studied in greater depth using other kinds of deep neural network architectures and datasets, they do coincide with the authors' findings and with the results of a recent research (Wilson, Roelofs, Stern, Srebro, & Recht, 2017).

Therefore, henceforth, the CNN $s_1\_c\_s_2\_c\_s_3\_c$ along with the SGD without momentum algorithm constitutes our proposed method for traffic sign classification, whose processing times for training and for testing one sample are 11.18 ± 0.02 μs and 4.28 ± 0.02 μs, respectively.

The following subsections describe the German and Belgian traffic sign datasets along with the classification results attained. The structure of each dataset is shown in Table 5 together with the overall recognition results. Note that these datasets are highly imbalanced, as can be observed in Fig. 6.

Table 5
European traffic sign classification datasets with their precision, recall and f1-score recognition results.

Dataset	Training images	Testing images	Classes
Germany	39,209	12,630	43
Belgium	4,533	2,562	62
	Precision (%)	Recall (%)	F1 score (%)
Germany	99.71	99.71	99.71
Belgium	98.95	98.87	98.86

Table 6
Recognition-rate accuracy of various methods on GTSRB.

Paper	Method	Accuracy (%)
Ours	Single CNN with 3 STNs	99.71
Jin et al. (2014)	HLSGD (20 CNNs ensemble)	99.65
Cireşan et al. (2012)	MCDNN (25 CNNs committee)	99.46
Yu et al. (2016)	GDBM	99.34
Stallkamp et al. (2011)	Human performance (best)	99.22
Jurisic et al. (2015)	OneCNN	99.11 ± 0.10

Table 7
Number of learneable parameters of our proposed CNN compared with that of previous state-of-the-art approaches.

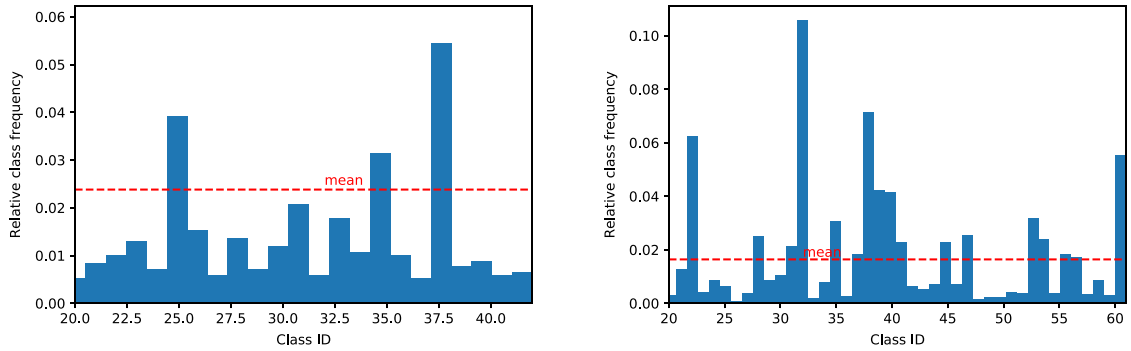
Paper	Data augment. or jittering	# trainable parameters	# ConvNets
Ours	No	14,629,801	1
Jin et al. (2014)	Yes	∼23 million	20 (ensemble)
Cireşan et al. (2012)	Yes	∼90 million	25 (committee)

4.1. GTSRB dataset results

The GSTRB dataset was introduced in Section 3.1. Our proposed CNN with three spatial transformer layers and SGD without momentum as the loss function optimiser achieves an accuracy of 99.71% at the 21st epoch (6 more than in the previous experiment). At the time of writing this paper, our method is top-1 ranked in the GTSRB and outperforms all previously published approaches (Table 6). In addition, the total number of parameters learnt by this CNN is 14,629,801, which is much lower than in other CNNs proposed for traffic sign recognition systems (Table 7), thereby leading to the further advantages of lower memory consumption, lower computational cost, and a simpler pipeline.

4.2. BTSC dataset results

The Belgian traffic sign classification dataset (BTSC) (Mathias et al., 2013) has 4533 training images and 2562 validation images split into 62 traffic sign types. In comparison with the GTSRB dataset, this dataset has different traffic sign pictograms, lighting conditions, occlusions, image resolutions, etc. Moreover, it contains categories that cluster different types of traffic signs (e.g. 50-speed-limit sign and 70-speed-limit sign), thereby raising the difficulty in the recognition task. By using the SGD without momentum loss optimiser algorithm and the CNN with three spatial



(a) Germany. (b) Belgium.

Fig. 6. European dataset category distribution.

Table 8
Recognition-rate accuracy of various methods on BTSC.

Paper	Method	Accuracy (%)
Yu et al. (2016)	GDBM	98.92
Ours	Single CNN with 3 STNs	98.87
Jurisic et al. (2015)	OneCNN	98.17 ± 0.22
Mathias et al. (2013)	INNLP+SRC(PI)	97.83

transformer layers, the model obtains an accuracy of 98.87% in the 13th epoch (Table 8).

5. Conclusions and future work

In this paper, a method for automatic fine-grained recognition of traffic signs is presented. The classification process is carried out by using a single CNN that alternates convolutional and spatial transformer modules. To find out the best CNN architecture, several empirical experiments are conducted in order to investigate the impact of multiple spatial transformer network configurations within the CNN, together with the effectiveness of four stochastic gradient descent optimisation algorithms. The CNN model outperforms all previous state-of-the-art methods and achieves a recognition rate accuracy of 99.71% in the GTSRB, and it is therefore currently top-1 ranked. Furthermore, our proposed approach needs no hand-crafted data augmentation and jittering used in prior work (Cireşan et al., 2012; Jin et al., 2014; Sermanet & LeCun, 2011). Moreover, there are fewer memory requirements and the network has a lower number of parameters to learn compared with existing methods since the use of several CNNs in a committee or in an ensemble is avoided.

Although our method is ranked in the top positions of the German and Belgian datasets, there have been several recent releases of publicly available traffic sign recognition datasets: these have not yet been tested since they are less established than previous datasets. Nevertheless, to the best of our knowledge, no other scientific paper analyses the use of several STNs and the comparison of stochastic gradient descent optimisers in the traffic sign classification problem domain. These experiments and their results can help other researchers to apply this new proposal to these new datasets.

Future work should study how to build a single deep neural network that could provide top-notch traffic sign recognition-rate accuracy in every country whose traffic sign pictographs are similar, which is the case of Europe, for which no particular dataset for any of the member countries is needed. Finally, we encourage researchers and companies to build traffic sign classifiers which are robust to those adversarial examples that could pose security concerns that may cause negative effects, such as in the use of self-driving cars, and consequently, may endanger other drivers and pedestrians alike.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness and FEDER R&D through the projects "Hermes-Smart Citizen" and "VICTORY" (Grants Nos.: TIN2013-46801-C4-1-R and TIN2017-82113-C2-1-R).

References

Barnes, N., Loy, G., & Shaw, D. (2010). The regular polygon detector. *Pattern Recognition*, *43*(3), 592–602.
Barnes, N., Zelinsky, A., & Fletcher, L. S. (2008). Real-time speed sign detection using the radial symmetry detector. *IEEE Transactions on Intelligent Transportation Systems*, *9*(2), 322–332.
Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*, *32*, 333–338.
Collobert, R., Kavukcuoglu, K., & Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, EPFL-CONF-192376.
De La Escalera, A., Moreno, L. E., Salichs, M. A., & Armingol, J. M. (1997). Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, *44*(6), 848–859.
Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, *12*, 2121–2159.
Escalera, A. D. L., Moreno, L., Salichs, M., & Armingol, J. (1997). Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, *44*(6), 848–859.
Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.
Gudigar, A., Chokkadi, S., Raghavendra, U., & Acharya, U. R. (2017). Local texture patterns for traffic sign recognition using higher order spectra. *Pattern Recognition Letters*, *94*, 202–210.
Huang, D.-S. (1996). *Systematic theory of neural networks for pattern recognition*. Publishing House of Electronic Industry of China, Beijing, Vol. 201.
Huang, D.-S. (1999). Radial basis probabilistic neural networks: Model and application. *International Journal of Pattern Recognition and Artificial Intelligence*, *13*(07), 1083–1101.
Huang, D.-S., & Du, J.-X. (2008). A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. *IEEE Transactions on Neural Networks*, *19*(12), 2099–2115.
Huang, G., Liu, Z., Weinberger, K. Q., & van der Maaten, L. (2016). Densely connected convolutional networks. arXiv preprint arXiv:1608.06993.
Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., & Pazhayampallil, J. et al., (2015). An empirical evaluation of deep learning on highway driving, arXiv preprint arXiv:1504.01716.
Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *Advances in neural information processing systems* (pp. 2017–2025).
Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision* (pp. 2146–2153).

Jin, J., Fu, K., & Zhang, C. (2014). Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *15*(5), 1991–2000.

Jurisic, F., Filkovic, I., & Kalafatic, Z. (2015). Multiple-dataset traffic sign classification with OneCNN. In *2015 3rd IAPR Asian conference on pattern recognition* (pp. 614–618).

Kaplan Berkaya, S., Gunduz, H., Ozsen, O., Akinlar, C., & Gunal, S. (2016). On circular traffic sign detection and recognition. *Expert Systems with Applications*, *48*, 67–75.

Kingma, D. P., & Ba, J. L. (2015). Adam: a method for stochastic optimization. In *International conference on learning representations 2015* (pp. 1–15).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Larsson, F., & Felsberg, M. (2011). Using fourier descriptors and spatial models for traffic sign recognition. In *Image analysis lecture notes in computer science, Vol. 11* (pp. 238–249).

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, J., Mei, X., Prokhorov, D., & Tao, D. (2017). Deep neural network for structural prediction and lane detection in traffic scene. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(3), 690–703.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37). Springer.

Loy, G., & Barnes, N. (2004). Fast shape-based road sign detection for a driver assistance system. In *2004 IEEE/RSJ international conference on intelligent robots and systems, 2004. Proceedings. Vol. 1* (pp. 70–75).

Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., & Lopez-Ferreras, F. (2007). Road-sign detection and recognition based on support vector machines. *IEEE Transactions on Intelligent Transportation Systems*, *8*, 264–278.

Mathias, M., Timofte, R., Benenson, R., & Van Gool, L. (2013). Traffic sign recognition How far are we from the solution? In *The 2013 international joint conference on neural networks* (pp. 1–8).

Mogelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1484–1497.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning* (pp. 807–814).

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate O (1/k2). In *Soviet mathematics doklady, Vol. 27* (pp. 372–376).

Oquab, M. (2017). stnbhwd.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, *4*(5), 1–17.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, *12*(1), 145–151.

Redmon, J., & Farhadi, A. (2016). YOLO9000: better, faster, stronger. arXiv preprint arXiv:1612.08242.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Salti, S., Petrelli, A., Tombari, F., Fioraio, N., & Di Stefano, L. (2015). Traffic sign detection via interest region extraction. *Pattern Recognition*, *48*(4), 1039–1049.

Scherer, D., Müller, A., & Behnke, S. (2010). *Lecture notes in computer science. Evaluation of pooling operations in convolutional architectures for object recognition* (pp. 92–101).

Sermanet, P., & LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *The 2011 international joint conference on neural networks* (pp. 2809–2813).

Shadeed, W., Abu-Al-Nadi, D., & Mismar, M. (2003). Road traffic sign detection in color images. In *10th IEEE international conference on electronics, circuits and systems, 2003. Proceedings of the 2003, Vol. 2* (pp. 890–893).

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The German traffic sign recognition benchmark: A multi-class classification competition. In *The 2011 international joint conference on neural networks* (pp. 1453–1460).

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, *32*, 323–332.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-resnet and the impact of residual connections on learning. In *AAAI* (pp. 4278–4284).

Tian, Y., Luo, P., Wang, X., & Tang, X. (2015). Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5079–5087).

Tieleman, T., & Hinton, G. (2012). Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*.

Timofte, R., & Van Gool, L. (2015). Iterative nearest neighbors. *Pattern Recognition*, *48*(1), 60–72.

Timofte, R., Zimmermann, K., & Van Gool, L. (2011). Multi-view traffic sign detection, recognition, and 3D localisation. *Machine Vision and Applications*, *25*(3), 633–647.

United Nations Economic Commission for Europe (1968). Convention on road signs and signals.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2017). The Marginal Value of Adaptive Gradient Methods in Machine Learning. arXiv preprint arXiv:1705.08292.

Youssef, A., Albani, D., Nardi, D., & Bloisi, D. D. (2016). Fast traffic sign recognition using color segmentation and deep convolutional networks. In *Advanced concepts for intelligent vision systems: 17th international conference, Lecce, Italy, October 24-27, 2016, proceedings* (pp. 205–216). Springer International Publishing.

Yu, Y., Li, J., Wen, C., Guan, H., Luo, H., & Wang, C. (2016). Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *113*, 106–123.

Zaklouta, F., Stanciulescu, B., & Hamdoun, O. (2011). Traffic sign classification using K-d trees and Random Forests. In *The 2011 international joint conference on neural networks* (pp. 2151–2155).

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-sign detection and classification in the wild. In *The IEEE conference on computer vision and pattern recognition* (pp. 2110–2118).

CAPÍTULO 4

EVALUATION OF DEEP NEURAL NETWORKS FOR TRAFFIC SIGN DETECTION SYSTEMS

Resumen

Los sistemas de detección de señales de tráfico constituyen un componente clave en aplicaciones actuales del mundo real, como la conducción autónoma, y la seguridad y asistencia del conductor. Este trabajo analiza el estado del arte de varios sistemas de detección de objetos (Faster R-CNN, R-FCN, SSD y YOLO V2) combinados con varios extractores de características (Resnet V1 50, Resnet V1 101, Inception V2, Inception Resnet V2, Mobilenet V1 y Darknet-19) desarrollados previamente por sus autores correspondientes. Nuestro objetivo es explorar las propiedades de estos modelos de detección de objetos modificándolos y adaptándolos específicamente al dominio del problema de la detección de señales de tráfico mediante la transferencia de conocimiento entre redes neuronales. En particular, varios modelos de detección de objetos disponibles públicamente que fueron entrenados previamente con el conjunto de imágenes COCO de Microsoft, se reajustan con el conjunto de imágenes del German Traffic Sign Detection Benchmark (GTSDB). La evaluación y compa-

ración de estos modelos incluyen métricas clave, como la precisión media promedio (mAP), el consumo de memoria, el tiempo de ejecución, el número de operaciones de punto flotante, el número de parámetros del modelo, y el efecto que tienen los tamaños de las imágenes de las señales de tráfico. Nuestros experimentos muestran que Faster R-CNN Inception Resnet V2 obtiene el mejor mAP, mientras que R-FCN Resnet 101 logra el mejor equilibrio entre precisión y tiempo de ejecución. Destacar igualmente los modelos YOLO V2 y SSD Mobilenet, ya que el primero logra resultados de precisión competitivos y es el segundo detector más rápido, mientras que el segundo es el modelo más rápido y ligero en términos de consumo de memoria, por lo que es una opción óptima para desplegarse como solución en dispositivos móviles y embebidos.

Neurocomputing 316 (2018) 332–344

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Evaluation of deep neural networks for traffic sign detection systems

Álvaro Arcos-García*, Juan A. Álvarez-García, Luis M. Soria-Morillo

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Sevilla 41012, Spain

ARTICLE INFO

Article history:
Received 25 March 2018
Revised 21 May 2018
Accepted 6 August 2018
Available online 11 August 2018

Communicated by Prof. Zidong Wang

Keywords:
Deep learning
Traffic sign detection
Convolutional neural network

ABSTRACT

Traffic sign detection systems constitute a key component in trending real-world applications, such as autonomous driving, and driver safety and assistance. This paper analyses the state-of-the-art of several object-detection systems (Faster R-CNN, R-FCN, SSD, and YOLO V2) combined with various feature extractors (Resnet V1 50, Resnet V1 101, Inception V2, Inception Resnet V2, Mobilenet V1, and Darknet-19) previously developed by their corresponding authors. We aim to explore the properties of these object-detection models which are modified and specifically adapted to the traffic sign detection problem domain by means of transfer learning. In particular, various publicly available object-detection models that were pre-trained on the Microsoft COCO dataset are fine-tuned on the German Traffic Sign Detection Benchmark dataset. The evaluation and comparison of these models include key metrics, such as the mean average precision (mAP), memory allocation, running time, number of floating point operations, number of parameters of the model, and the effect of traffic sign image sizes. Our findings show that Faster R-CNN Inception Resnet V2 obtains the best mAP, while R-FCN Resnet 101 strikes the best trade-off between accuracy and execution time. YOLO V2 and SSD Mobilenet merit a special mention, in that the former achieves competitive accuracy results and is the second fastest detector, while the latter, is the fastest and the lightest model in terms of memory consumption, making it an optimal choice for deployment in mobile and embedded devices.

1. Introduction

Traffic sign recognition systems (TSRS) form an important component of Advanced Driver-Assistance Systems (ADAS) and are essential in many real-world applications, such as autonomous driving, traffic surveillance, driver safety and assistance, road network maintenance, and analysis of traffic scenes. A TSRS normally concerns two related subjects: traffic sign detection (TSD) and traffic sign recognition (TSR). The former focuses on the localisation of the target in a frame, while the latter performs a fine-grained classification to identify the type of the detected target [1,2].

Traffic signs constitute a fundamental asset within the road network because their aim is to be easily noticeable by pedestrians and drivers in order to warn and guide them both day and night. The fact that signs are designed to be unique and to have distinguishable features, such as simple shapes and uniform colours, implies that their detection and recognition is a constrained problem. Nevertheless, the development of a robust real-time TSRS still presents a challenging task due to the latency in the testing time, which is crucial in making decisions based on the environment and real-world variability, such as scale variations, bad viewpoints, occlusions, and light conditions. Any TSRS must cope well with such issues.

An ADAS relies on LiDAR, onboard RGB cameras, GPS, and IMU sensors. Although traffic signs are normally geo-located and included in navigation maps, they are sometimes replaced or included before the map is updated. The fusion of complementary information acquired from both LiDAR and RGB cameras is a common approach used in TSRS [3,4]. However, 3D point cloud processing is computationally expensive and the calibration of the sensors and cameras has to be precise since errors of measurement in the order of centimeters may lead to deficient performance.

In recent years, most of the state-of-the-art object-detection algorithms, such as Faster R-CNN [5], R-FCN [6], SSD [7], and YOLO [8], have used convolutional neural networks (CNNs) and can be deployed in mobile devices and consumer products. In order to decide which detector best suits a certain application, not only are standard accuracy metrics important, such as the mean average precision (mAP), but other factors, such as memory consumption and running times, also play a critical role. For instance, autonomous vehicles require good detection accuracy and real-time performance, while mobile devices require lightweight model architectures with low memory usage. In the literature, those

* Corresponding author.
E-mail addresses: aarcos1@us.es (Á. Arcos-García), jaalvarez@us.es (J.A. Álvarez-García), lsoria@us.es (L.M. Soria-Morillo).

https://doi.org/10.1016/j.neucom.2018.08.009
0925-2312/

detectors are commonly evaluated in object-detection challenges, such as Imagenet [9], PASCAL VOC [10], and Microsoft COCO [11], whose corresponding datasets contain numerous images with common objects, such as cars, planes, people, and bicycles, whereby only accuracy results are reported. However, recent work evaluated the performance of these modern detectors and reported the key metrics, using the Microsoft COCO dataset [12]. Since many of the leading state-of-the-art object-detection approaches have converged on a common methodology that consists of a single CNN that uses sliding-window-style predictions and is trained with a mixed regression and classification objective, the authors implement meta-architectures of Faster R-CNN, R-FCN, and SSD combined with various feature extractors, in order to compare a large number of detection systems in a unified manner.

This paper analyses and compares eight CNN models for object detection that have been previously developed by their corresponding authors and pre-trained on the Microsoft COCO dataset. We fine-tune them on the German Traffic Sign Detection Benchmark dataset (GTSDB) [13] in order to perform traffic sign detection. Considering that the training process of deep CNNs using a very large dataset (e.g. the COCO dataset) requires High Performance Computing (HPC) resources, such as multiple GPUs, and several weeks of continuous training time, we perform transfer learning through fine-tuning to deal with such issues, which consists of reusing the weights learnt by a trained network on another related network [14]. Evaluated detection models are combinations of meta-architectures (Faster R-CNN, R-FCN, SSD, and YOLO V2) and feature extractors (Resnet V1 50, Resnet V1 101, Inception V2, Inception Resnet V2, Mobilenet V1, and Darknet-19). Such models, pre-trained on the COCO dataset, are publicly available.[1,2]

To the best of our knowledge, no other scientific paper analyses several object detectors based on deep learning that are specifically adapted to the domain of the traffic sign detection problem, while evaluating multiple important factors, such as mAP, inference execution time, and memory consumption. The main contributions of this paper are as follows: (1) Presentation of a brief survey of modern object-detection algorithms based on CNNs, namely Faster R-CNN, R-FCN, SSD, and YOLO. (2) Analysis and evaluation of several state-of-the-art object detectors tuned especially for the traffic sign detection task. The evaluation of these models includes key metrics, such as the mAP, memory usage, running time, number of floating point operations (FLOPs), number of parameters of the model, and the effect of traffic sign image sizes. (3) Comparisons and experiments that are made publicly available so that researchers and practitioners can improve their knowledge and fine-tune new models for their comparison with our experimentation. (4) Findings that show that R-FCN strikes the best trade-off between speed and accuracy, SSD models are weak at detecting small traffic signs, and that Mobilenet is the best architecture suited for mobile and embedded devices.

The rest of the paper is organised as follows. Section 2 reviews related work of traffic sign detection systems. Methodology and experiments conducted to analyse several state-of-the-art CNN architectures for object detection are explained in Section 3. In Section 4, the traffic sign detection results obtained are analysed, compared and discussed. Finally, conclusions are drawn and further work is proposed in Section 5.

2. Related work

State-of-the-art research in this field is analysed from two points of view: firstly, traffic sign detection solutions; secondly, deep neural network architectures for object detection.

[1] https://github.com/tensorflow/models/blob/master/research/object_detection
[2] https://pjreddie.com/darknet/yolo

2.1. Traffic sign detection

Various approaches have been studied for traffic sign detection systems. In [15], a detector composed of two modules is proposed. The former exploits the common properties of sign borders and extracts regions of interest (ROI). The latter performs finer validations over the ROIs and detects traffic signs using a combination of Histograms of Oriented Gradients (HOG) and Support Vector Machines (SVM). A sliding-window detector approach is proposed in [16], where integral channel features classifiers are applied along with the search for traffic signs on different scales and aspect ratios. Wang et al. [17] proposed the winner method for the prohibitory and mandatory signs in the German Traffic Sign Detection Benchmark (GTSDB) [13] challenge. Their system combines a coarse filtering module based on HOG and Linear Discriminant Analysis (LDA) classification on small sliding windows, and a fine filtering module, which includes HOG of larger windows and a SVM classifier. The previous methods are based on the sliding-window schema and feature extraction, which is time-consuming and complex and hence they are not useful for real-time object detection. Recently, Zang et al. [18] combine a local binary pattern (LBP) feature detector with an AdaBoost classifier [19] in order to extract ROIs for coarse selection followed by cascaded CNNs to reduce negative samples of ROI for traffic sign recognition. In 2016, Zhu et al. [20] develop a method to detect and recognise traffic signs based on proposals by the guidance of fully convolutional network. They extent the R-CNN by using an object proposal method, EdgeBox [21] and achieve state-of-the-art results on Swedish Traffic Signs Dataset [22]. Additionally in 2016, Aghdam et al. [23] propose a method that implements the multi-scale sliding window technique within a CNN using dilated convolutions. Dilated convolutions (also known as atrous convolutions) support the exponential expansion of the receptive field without any loss of resolution or coverage and hence they enlarge the field of view of convolutional filters to incorporate a larger context without increasing the amount of computation or the number of parameters [24]. Such an approach locates traffic signs on the GTSDB high-resolution images with an average precision of 99.89% and runs at 37.72fps. An overview of these revised TSD systems, evaluated on GTSDB, is shown in Table 1.

2.2. Convolutional neural networks for object detection

Since 2013, CNNs, which are able to learn a hierarchy of features by building high-level features from low-level features, have become the standard for object-detection tasks. Examples include are OverFeat [25], which produces bounding boxes and scores using CNNs in a sliding-window fashion, and R-CNN [26], which follows a multi-stage pipeline where object region proposals are extracted from the input image by means of Selective Search [27], whereby feature maps are then computed with a CNN for every region proposal, and finally bounding box regressors and SVM classifiers are applied. R-CNN is expensive both in time and memory because it executes a CNN forward-pass for each object proposal without sharing computation.

To face such issues, Spatial Pyramid Pooling networks (SPPnets) [28] were proposed to improve R-CNN efficiency by sharing computation. SPPnet computes the feature maps from the entire input image only once, and then pools features in sub-images of arbitrary size to generate fixed-length representations to train the detectors. Although the repeated calculation of convolutional feature maps is obviated in SSPnet, it still requires training in a multi-stage pipeline since the fixed-length feature vectors produced by multiple SPP layers are further passed on to fully-connected layers and then, on top of these, bounding box regressors and SVMs are applied. Therefore, the whole process is still slow. Moreover, SSPnet

Table 1
Evaluation results, inference time and hardware utilised in various TSD systems tested on the GTSDB. P refers to prohibitory class, D to danger and M to mandatory.

Paper	Evaluation (%)				Inference time (FPS)	Hardware	
	Metric	P	D	M		CPU	GPU
Liang et al.(2013) [15]	AUC	100	98.85	92	1–2.5	Intel 4-core 3.7 GHz	–
Mathias et al. (2013) [16]	AUC	100	100	96.98	2.5	Intel Core i7 870 3.6 GHz	NVIDIA GTX 470
Wang et al. (2013) [17]	AUC	100	99.91	100	0.85	Intel Core i3 3.3 GHz	–
Zang et al. (2016) [18]	AUC	99.45	98.33	96.5	*	Intel Core 2 Duo 2.2 GHz	–
Aghdam et al. (2016) [23]	AP		99.89		37.72	–	NVIDIA GTX 980

* Time of the full process is not included.

introduces a new problem since parameters below the SPP layer cannot be updated while training.

The more recent Fast R-CNN [29] proposes a new training algorithm that provides solutions to fix the disadvantages of R-CNN and SPPnet, while improving on their speed and accuracy by sharing computation, and by training in a single-stage using a multi-task loss and reducing memory consumption. Instead of applying multiple SPP layers as in SSPnets, Fast R-CNN uses a single-level SPP layer, which is called the RoI Pooling layer. Furthermore, the multi-task loss is calculated on top of the network where bounding box regressors and softmax classifiers are applied. The training of the layers below the RoI Pooling layer is possible thanks to these changes, thereby overcoming the original problem of SPPnets. Although SPPnet and Fast-RCNN had reduced the running time of these detection networks, there was a bottleneck exposed in the generation of regions of interest from a proposal method.

In Faster R-CNN [5], in order to overcome such a bottleneck, authors replaced the use of Selective Search with a Region Proposal Network (RPN) that shares convolutional feature maps with the detection network, thus enabling nearly cost-free region proposals. Similar to Faster R-CNN, the Region-based Fully Convolutional Networks (R-FCN) [6] approach applies position-sensitive score maps along with a fully-convolutional region-based detector with shared computation that has no need for the per-region subnetwork to be executed hundreds of times per image. Other approaches, such as Single Shot MultiBox Detector (SSD) [7] and YOLO (You Only Look Once) [8], encapsulate all the computation in a single fully-convolutional neural network instead of having a sequential pipeline of region proposals and object classification. This ability leads to a much faster object detector.

3. Experimentation

The following subsections describe the dataset and the specific configuration used in several CNNs that are fine-tuned for traffic sign detection. Following [12], our experimental setup is composed of four meta-architectures (Faster R-CNN, R-FCN, SSD, and YOLO V2) and six convolutional feature extractors (Resnet V1 50, Resnet V1 101, Inception V2, Inception Resnet V2, Mobilenet V1, and Darknet-19). The feature extractors considered are all well-known convolutional neural networks for image classification that are applied to the input image to obtain high-level features.

Due to time restrictions and computational costs, all experiments presented in this paper use publicly available object-detection models that were pre-trained on the Microsoft COCO dataset [11]. By means of transfer learning [30], we fine-tune these models with the GTSDB dataset in order to detect and classify traffic sign superclasses based on their shapes and colours: mandatory, prohibitory, and danger. At the time of writing this paper, all pre-trained models available at the official repositories of Tensorflow Object Detection API [12] and YOLO [8] were used in our experimental setup. The combinations of architectures and feature extractors studied in this work are presented in Table 2. It can be observed that not all possible combinations have been explored. The reason is that each feature extractor must be tailored for use within a meta-architecture. These not trivial adjustments need several experiments and weeks of training, and hence only pre-trained combinations have been selected.

Table 2
Feature extractors vs. architectures. Combinations of CNN architectures and feature extractors evaluated in this paper.

	Faster R-CNN	R-FCN	SSD	YOLO V2
Resnet V1 50	✓			
Resnet V1 101	✓	✓		
Inception V2	✓		✓	
Inception Resnet V2	✓			
Mobilenet V1			✓	
Darknet-19				✓

3.1. Datasets

Several publicly available traffic sign datasets have been gathered in countries such as the United States [31], Belgium [32], Germany [13], Croatia [33], Italy [34], Sweden [22], and China [35].

This paper focuses its experimentation on the German Traffic Sign Detection Benchmark (GTSDB) [13] dataset. There are multiple reasons for choosing this dataset over the others, including the fact that it is highly accepted and is widely used for comparing traffic sign detection approaches in the literature. Moreover, its authors and the organisation behind them held a public challenge, whereby scientists from different fields contributed their results and tested the GSTDB dataset. Nowadays, a GTSDB website is maintained where submissions of results are still accepted, processed and shown in a leaderboard. Such ranking helps to reveal which state-of-the-art methodologies are utilised for the task of traffic sign detection, although their processing times are not considered. Last but not least, the GTSRB dataset contains natural traffic scenes recorded in various types of roads (highway, rural, urban) during the daytime and at twilight, and numerous weather conditions are featured. This dataset is composed of 900 full images containing 1206 traffic signs that are split into a training set of 600 images (846 traffic signs) and a testing set with 300 images (360 traffic signs). Each of these images contains zero, one, or multiple traffic signs which normally suffer from differences in orientation, light conditions, or occlusions. Signs are grouped in four categories namely mandatory, prohibitory, danger, and other, however, signs labelled as other remain in minority and are not relevant to the challenge itself, and hence are discarded. Consequently, the training set contains 396 prohibitory (59.5%), 114 (17.1%) mandatory and 156 (23.4%) danger traffic sign samples while the testing set comprises 161 prohibitory, 49 mandatory and 63 danger traffic sign images. The following deep neural networks for traffic sign detection are trained and evaluated using this dataset. Fig. 1 shows some images from this dataset. The following sections and subsections describe each meta-architecture used and its feature extractors.

Fig. 1. Example images from GTSDB dataset.

3.2. Meta-architectures for object detection

In this subsection, the main features of each meta-architecture (Faster R-CNN, R-FCN, SSD, and YOLO V2) are summarised.

3.2.1. Faster R-CNN

As mentioned in Section 2, Faster R-CNN [5] introduces a Region Proposal Network (RPN), which is a fully convolutional neural network that simultaneously predicts object bounding boxes and objectness scores. It makes the model completely trainable end-to-end since full-image convolutional feature maps are shared with the detection network. Region proposals are generated in a sliding-window fashion, sliding a small network over the output feature map of the latest convolutional layer. The RPN predicts multiple region proposals at each sliding-window location, where k is the maximum number of possible proposals for each location. The k proposals are parameterised relative to k reference boxes called anchors. Each of these anchor boxes are associated with an aspect and scale ratio, and centred at the sliding-window location. In order to reduce redundancy of overlapping RPN proposals, non-maximum suppression (NMS) algorithm is first performed on the proposal regions based on their objectness scores. The NMS algorithm is responsible for merging multiple detections that belong to the same object. Only the $top-N$ ranked proposal regions are then forwarded to the detection network, which finally regresses bounding boxes and classifies each of them in a determined object class.

During experimentation, the number of region proposals to be sent to the box classifier is set to 300 as this is the number of boxes used by the authors in their corresponding papers. Moreover, each feature extractor is trained on images scaled to 600 pixels on their shortest edge using a SGD with momentum (set to 0.9) as the loss-function optimiser [36] along with batch sizes of 1. The initial learning rate is set to 0.0003 and is manually reduced by a factor of 10 twice: after 900,000 iterations and 1,200,000 iterations.

3.2.2. R-FCN

Region-based Fully Convolutional Networks (R-FCN) [6] take the architecture of Faster R-CNN but with only convolutional neural networks. That is, the R-FCN approach applies a fully convolutional region-based detector whose computation is shared across the entire image, thereby obviating the need for the computation of per-region subnetwork to be executed hundreds of times per image. To this end, authors propose position-sensitive score maps to address a dilemma between translation-invariance in image classification (where the shift of an object inside an image should be indiscriminate), and translation-variance in object detection (where the detection task needs meaningful localisation representations for the evaluation of how the candidate box overlaps the object). Therefore, R-FCN adopts a sequential two-stage pipeline of region proposal and region classification where candidate regions are extracted by a fully convolutional RPN.

In the same way as for Faster R-CNN, the training configuration as well as the hyper-parameter tuning is exactly the same as was described in Section 3.2.1 above.

3.2.3. SSD

In comparison with Faster R-CNN and R-FCN architectures, SSD [7] encapsulates all computation in a single feed-forward convolutional neural network to directly infer box offsets and object category scores. Consequently, a stage of bounding box proposal generation and subsequent feature or pixel resampling is not required. SSD uses a set of default boxes (also known as anchors or anchor boxes) that are hand-picked by the developer who has to previously observe the size of the objects to be detected. These default boxes aim to discretise the output space of bounding boxes over different scales and aspect ratios per feature map location. That is, at each feature map cell, SSD predicts the offsets relative to the anchor shapes in the cell, as well as the category scores that indicate the presence of an object class instance in each of those anchors.

Moreover, to handle objects of multiple sizes, SSD combines predictions from feature maps of different resolutions. The early network layers of an SSD model are based on a standard architecture used for high-quality image classification. An auxiliary structure is then added to the network in order to produce multi-scale feature maps for detection purposes. Such a structure is composed of convolutional feature layers whose aim is to decrease the size of these feature maps progressively and allow predictions of detections on multiple scales.

For experimentation, unlike Faster R-CNN and R-FCN, SSD models are trained using RMSprop [37] with a momentum of 0.9 as the loss-function optimiser and batch sizes of 32. The base learning rate is set to 0.004 and is exponentially decayed by a factor of 0.95 for each 800,000 iterations. As regards input image sizes, they are resized to a fixed shape of 300×300 pixels.

3.2.4. YOLO V2

YOLO V2 [8] is inspired by the RPN of Faster R-CNN, which uses hand-picked anchor boxes to predict bounding boxes based on the offsets to these anchors at every location in a feature map. However, on one hand, the YOLO V2 approach runs k-means clustering on the training-set bounding boxes using a custom distance metric (Eq. (1)) in order to find good anchor boxes instead of choosing them by hand. Picking better anchor boxes makes it easier for the network to learn to predict good detection. On the other hand, in

order to prevent any anchor box ending up at any point in the image, it predicts the width and height of the box as offsets from cluster centroids and location coordinates relative to the location of the grid cell, by applying a logistic activation to constrain the predictions of the network to fall between 0 and 1.

$$d(box, centroid) = 1 - IoU(box, centroid) \quad (1)$$

The classification model that is used as the base of YOLO V2 is called Darknet-19. Furthermore, YOLO V2 uses batch normalisation, which helps to regularise the model and leads to notable improvements in convergence while stabilising the model [38]. After training, the network is modified and fine-tuned for object detection as described in Section 3.3.5.

In order to improve detection scores, standard data augmentation, such as rotations, random crops and exposure, hue and saturation shifts, are performed along with multi-scale training, which re-sizes the input image size every few iterations, thereby forcing the network to learn to predict detections at different resolutions.

In the same way as for SSD, the loss-function optimiser applied to train the model is RMSprop with a momentum of 0.9 and batch sizes of 64. Moreover, the input image size is 608 × 608 pixels, and the initial learning rate is set to 0.001, which is decayed by a factor of 10 at steps 400,000 and 450,000.

3.3. Feature extractors

We adopt well-known convolutional neural networks for image classification that will be used as feature extractors to obtain high-level features from input images: Resnet V1 50, Resnet V1 101, Inception V2, Inception Resnet V2, Mobilenet V1, and Darknet-19.

3.3.1. Resnet V1 50 and Resnet V1 101

Resnet V1 101 and Resnet V1 50 are deep residual networks [39] that have succeeded in many challenges, such as ILSVRC, and COCO 2015 (detection, segmentation and classification). To be used as feature extractors of Faster R-CNN and R-FCN meta-architectures, these networks are are split into two stages. The former performs the extraction of RPN features and the latter extracts box classifier features.

Both of these feature extractors are built with four residual blocks: the first three (namely *conv2_x*, *conv3_x*, and *conv4_x* in the original paper) extract RPN features, while the last layer of *conv4_x* is used for predicting region proposals. Additionally, box classifier features are extracted by the last layer of the fourth residual block (*conv5_x*).

3.3.2. Inception V2

Inception V2 [40] sets the state-of-the-art in the ILSVRC2014 detection and classification challenges. Inception networks make use of Inception units that are able to increase the depth and width of a network without increasing its computational cost.

On one hand, when this feature extractor is used in combination with Faster R-CNN meta-architecture, RPN feature maps are extracted from the *Mixed_4e* layer and proposal classifier features from the *Mixed_5c* layer. These layers are called respectively *inception*(4*e*) and *inception*(5*b*) in the network architecture described in [40].

On the other hand, when SDD is applied as a meta-architecture, the feature extraction of region proposals is not required in SSD (as was mentioned in Section 3.2.3), and hence Inception V2 is not split, but instead the whole network model is adopted as the main feature extractor. However, auxiliary convolutional feature maps on multiple scales are needed. The topmost convolutional feature map and a high resolution feature map at a lower level are selected. A sequence of four convolutional layers with batch normalisation and depths 512, 256, 256, and 128, is then appended to the previously selected layers to perform the prediction task. Each of these additional layers decay the spatial resolution of feature maps by a factor of 2. For Inception V2, multi-resolution feature maps are generated by the layers *Mixed_4c* and *Mixed_5c*.

3.3.3. Inception Resnet V2

In the case of Inception Resnet V2 [41], the computation efficiency of Inception units are combined with the optimisation benefits conferred by residual connections. This feature extractor is only combined with Faster R-CNN meta-architecture in our experiments and hence can be split into two stages. On one hand, RPN features are extracted from the *Mixed_6a* layer including its associated residual layers (17 × 17 grid module, known as Inception-ResNet-B in [41]). On the other hand, box classifier features are obtained using the layers located immediately after the Inception-ResNet-B module up to the convolutional layer *Conv2d_7b_1x1*, which follows the 8 × 8 grid module named Inception-ResNet-C in [41]. This feature extractor is operated with dilated convolutions so that the effective output stride size is 8 pixels.

3.3.4. Mobilenet V1

The Mobilenet V1 [42] model is designed for efficient inference in mobile vision applications thanks to the use of depthwise separable convolutions that reduce both the number of parameters and the computational cost. In fact, Mobilenet V1 achieves the same level of accuracy as VGG-16[43] on Imagenet with only 1/30 of the model size and computational cost.

This feature extractor is used in combination with SDD meta-architecture in our experiments, for that reason, its network architecture is not split and auxiliary convolutional feature maps at multiple scales are needed. Analogously to the modifications performed when using Inception V2 with SSD (Section 3.3.2), multi-resolution feature maps are generated by the layers *conv_11* and *conv_13*, and four additional convolutional layers are then appended with decaying resolution and depths 512, 256, 256, and 128, respectively.

3.3.5. Darknet-19

As described in Section 3.2.4, the original object-classification model Darknet-19, which acts as a feature extractor, is modified to perform object detection. Darknet-19 is similar to the VGG [43] model architecture since it doubles the number of feature maps after every pooling layer and uses chiefly 3×3 kernels. Moreover, it applies a global average pooling to make predictions together with 1×1 kernels to reduce space dimensionality between 3×3 convolutions [44].

This model is first trained on Imagenet using images of 224 × 224 pixels. The model is then fine-tuned at a larger image size for a few epochs. This gives the network time to adjust its filters to work better on higher resolution inputs. Finally, the network is modified and fine-tuned for detection by removing the last convolutional layer and replacing it by adding on three 3 × 3 convolutional layers with 1024 filters each followed by a final 1 × 1 convolutional layer with the number of outputs needed for the detection task.

The result is the YOLO V2 model architecture shown in Table 3. The output feature maps of the last convolutional layer depend on several factors including the number of predicted bounding boxes at each cell *PredB*, which corresponds to the number of anchor boxes, the number of coordinates *CoorB*, and the number of different classes *ClassB*. We set $CoorB = 5$ as the network predicts the centre coordinates, width, height and confidence per each bounding box resulting in $(ClassB + CoorB) * PredB = (ClassB + 5) * PredB$ output feature maps. For this experiment, we set 5 anchor boxes

Table 3
YOLO V2 network architecture.

Layer	Type	# Maps & neurons	Kernel size/stride
0	Input	3 m. of 608 × 608 n.	
1	Conv	32 m. of 608 × 608 n.	3 × 3/1
2	Max-Pool	32 m. of 304 × 304 n.	2 × 2/2
3	Conv	64 m. of 304 × 304 n.	3 × 3/1
4	Max-Pool	64 m. of 152 × 152 n.	2 × 2/2
5	Conv	128 m. of 152 × 152 n.	3 × 3/1
6	Conv	64 m. of 152 × 152 n.	1 × 1/1
7	Conv	128 m. of 152 × 152 n.	3 × 3/1
8	Max-Pool	128 m. of 76 × 76 n.	2 × 2/2
9	Conv	256 m. of 76 × 76 n.	3 × 3/1
10	Conv	128 m. of 76 × 76 n.	1 × 1/1
11	Conv	256 m. of 76 × 76 n.	3 × 3/1
12	Max-Pool	256 m. of 38 × 38 n.	2 × 2/2
13	Conv	512 m. of 38 × 38 n.	3 × 3/1
14	Conv	256 m. of 38 × 38 n.	1 × 1/1
15	Conv	512 m. of 38 × 38 n.	3 × 3/1
16	Conv	256 m. of 38 × 38 n.	1 × 1/1
17	Conv	512 m. of 38 × 38 n.	3 × 3/1
18	Max-Pool	512 m. of 19 × 19 n.	2 × 2/2
19	Conv	1024 m. of 19 × 19 n.	3 × 3/1
20	Conv	512 m. of 19 × 19 n.	1 × 1/1
21	Conv	1024 m. of 19 × 19 n.	3 × 3/1
22	Conv	512 m. of 19 × 19 n.	1 × 1/1
23	Conv	1024 m. of 19 × 19 n.	3 × 3/1
24	Conv	1024 m. of 19 × 19 n.	3 × 3/1
25	Conv	1024 m. of 19 × 19 n.	3 × 3/1
26	Route(17)	512 m. of 38 × 38 n.	
27	Reorg	2048 m. of 19 × 19 n.	-/2
28	Concat(25,27)	3072 m. of 19 × 19 n.	
29	Conv	1024 m. of 19 × 19 n.	3 × 3/1
30	Conv	40 m. of 19 × 19 n.	1 × 1/1

computed through the k-means clustering algorithm, and hence the number of filters of the last convolutional layer is 40.

4. Results

In this section we present the performance of the traffic sign detector experiments described in Section 3. The analysis of each of these experiments includes multiple measures, such as accuracy, number of parameters, floating point operations (FLOPs), memory consumption, and processing time. The models are trained and evaluated on a computer built with an Intel Core i7-4770 CPU, 16 GB of RAM and a NVIDIA Titan Xp discrete GPU, which has 3840 CUDA cores and 12 GB of RAM. As development tools, we used Darknet,[3] Darkflow[4] and the Tensorflow Object Detection API [12].

Timings are comprised of both GPU and CPU execution times, and post-processing tasks, such as NMS, are also included. Both execution times and memory demand are reported for a batch size of one and they are averaged over 300 images (GTSDB testing set). The Tensorflow profiler tool[5] was employed to compute these measures as well as the number of parameters and floating point operations (multiply-adds). Our timings are comparable to each other, however, they may not be directly comparable to other reported timing results in the literature since major differences could exist within the computer used, such as software drivers, hardware, framework, and batch size. Nevertheless, factors, such as the total memory allocation of the models during inference, the number of parameters, and the floating point operations, constitute platform-independent measures.

[3] http://pjreddie.com/darknet/ (accessed 11.09.2017)
[4] https://github.com/thtrieu/darkflow (accessed 11.09.2017)
[5] https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profiler (accessed 27.10.2017)

4.1. Accuracy evaluation measure

The mean Average Precision (mAP) quantitative measure from PASCAL VOC 2010 [10] is used to evaluate the performance of the proposed traffic sign detector. First, the interpolated Average Precision (AP), which tracks the precision/recall curve, is computed by setting the precision for recall r to the maximum precision obtained for any recall $r' \geq r$ (Eq. (2)), where $p(r')$ is the measured precision at recall r'. The AP measure can then be calculated as the area under this curve by numerical integration that is approximated by the sum of the precision at every k where the recall changes, multiplied by the change in recall $\Delta r(k)$ (Eq. (3)), where N is the total number of points where recall changes. Finally, the mAP measure is calculated by taking the average of the APs of all the classes.

$$p(r) = \max_{r':r' \geq r} p(r') \tag{2}$$

$$AP = \sum_{k=1}^{N} p(k)\Delta r(k) \tag{3}$$

In order to determine true and false positive predicted bounding boxes B_p, their respective intersection over union (IoU) with the ground truth bounding boxes B_{gt} are computed. IoU is defined as the area of overlap $B_{gt} \cap B_p$ divided by the area of union $B_{gt} \cup B_p$ (Eq. (4)). A prediction is correct when its IoU is greater than 0.5 and it is a false positive otherwise. Moreover, ground truth objects with no matching detection are false negatives and multiple detections on the same traffic sign in an image are considered as false positives.

$$IoU = \frac{area(B_{gt} \cap B_p)}{area(B_{gt} \cup B_p)} = \frac{area(B_{gt} \cap B_p)}{area(B_{gt}) + area(B_p) - area(B_{gt} \cap B_p)} \tag{4}$$

4.2. Analyses

Detailed accuracy results per traffic sign superclass are presented in Table 4 along with precision, recall, average precision, and average IoU attained by each detector. On one hand, the worst AP belongs to the mandatory traffic sign category in almost all models, and noticeable differences exist between the AP of the other classes, especially in detectors with lightweight feature extractors, such as SSD variants and YOLO V2. On the other hand, every evaluated model obtains the best AP in its detection of prohibitory traffic sign images. Faster R-CNN Inception Resnet V2 even achieves an accuracy of 100% in this category. These results are highly correlated with the number of training traffic sign samples that are included in the GTSDB dataset described in Section 3.1, where 59.5% are prohibitory and only 17.1% are mandatory. Table 5 includes the FPS, Megabytes of memory, Gigaflops, and millions of parameters of each model sorted by its mAP.

The execution time is really a critical factor for real-time TSD systems. The overall mAP achieved by each model configuration together with its processing time are drawn in Fig. 2. Three groups are observed. The first group is comprised of the fastest models, YOLO V2 and SSD, which do not perform region proposal generation. YOLO V2 outperforms the SSD models in terms of mAP, although they do have similar running times. SSD Mobilenet is the fastest of all the models, with an execution time of 15.14ms per image (66fps) although its accuracy is slightly worse than that of SSD Inception V2. The second cluster is composed of the Faster R-CNN models with lightweight feature extractors and R-FCN Resnet 101. These models are more accurate and require approximately 100 ms per image on average. In fact, the accuracies obtained by R-FCN and Faster R-CNN when the feature extractor is a Resnet 101

Table 4
GTSDB accuracy results (in %) as attained by each traffic sign detector model. Average IoU only takes IoU values of true positive bounding boxes.

Model	Class	Avg. IoU	Precision	Recall	AP
Faster R-CNN Resnet 50	Prohibitory	82.52	91.38	98.75	98.62
	Mandatory	81.21	70.00	85.71	85.15
	Danger	85.07	79.45	92.06	90.78
Faster R-CNN Resnet 101	Prohibitory	87.29	90.29	98.14	98.13
	Mandatory	85.58	67.65	93.88	93.46
	Danger	87.05	85.51	93.65	93.64
Faster R-CNN Inception V2	Prohibitory	82.73	81.22	99.38	99.36
	Mandatory	79.66	62.50	81.63	80.47
	Danger	85.62	81.69	92.06	92.03
Faster R-CNN Inception Resnet V2	Prohibitory	91.37	96.99	100	100
	Mandatory	89.16	79.31	93.88	93.66
	Danger	90.11	92.19	93.65	93.65
R-FCN Resnet 101	Prohibitory	87.93	84.66	99.38	99.37
	Mandatory	85.37	76.67	93.88	92.58
	Danger	86.95	86.76	93.65	93.52
SSD Inception V2	Prohibitory	81.76	96.95	78.88	78.77
	Mandatory	80.85	90.00	55.10	54.46
	Danger	85.76	93.18	65.08	65.05
SSD Mobilenet	Prohibitory	80.49	92.50	68.94	67.03
	Mandatory	78.51	89.65	53.06	52.01
	Danger	81.11	79.63	68.25	65.85
YOLO V2	Prohibitory	73.96	92.31	89.44	88.73
	Mandatory	74.66	79.07	69.39	65.70
	Danger	75.82	94.55	82.54	82.06

Table 5
Models' properties ordered by mAP.

Model	mAP	FPS	Memory (MB)	GigaFLOPS	Parameters (10^6)
Faster R-CNN Inception Resnet V2	95.77	2.26	18250.45	1837.54	59.41
R-FCN Resnet 101	95.15	11.70	3509.75	269.90	64.59
Faster R-CNN Resnet 101	95.08	8.11	6134.71	625.78	62.38
Faster R-CNN Resnet 50	91.52	9.61	5256.45	533.58	43.34
Faster R-CNN Inception V2	90.62	17.08	2175.21	120.62	12.89
YOLO V2	78.83	46.55	1318.11	62.78	50.59
SSD Inception V2	66.10	42.12	284.51	7.59	13.47
SSD Mobilenet	61.64	66.03	94.70	2.30	5.57

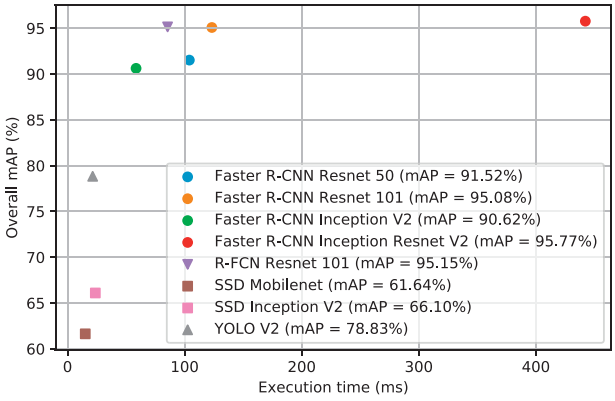


Fig. 2. Accuracy vs. execution time.

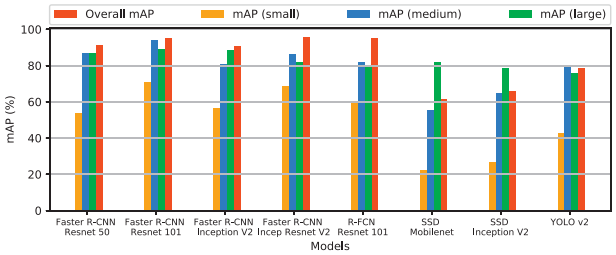


Fig. 3. Accuracy classified by traffic sign size for 8 different detectors.

network, are very close to the Faster R-CNN Inception Resnet V2 model (third group), which attains the best mAP: 95.77%. However, it is by far the slowest model due to its processing time, which is almost half a second. Consequently, the R-FCN Resnet 101 model strikes the best balance between accuracy and speed among the model configurations studied, since it achieves an mAP of 95.15% and takes 85.45ms per image (11.7fps). A faster option with still good accuracy is that of YOLO V2, which runs at 21.48 ms (46.55 fps).

Additionally, we notice that traffic sign image sizes have negative effects on accuracy. Ground truth traffic sign samples that belong to the validation set are divided into three groups regarding their width. The first group contains 89 images, whose width is in the range [0,32). The second group has 93 samples, and their width is included in the range [32,46). The third group clusters 91 images, which width is greater than 45 pixels. All detectors perform better on larger traffic sign images, as can be seen in Fig. 3. One possible reason that explains this fact is that the initial convolutional weights of the networks evaluated were pre-trained on the Microsoft COCO dataset and most of its images have large objects in the centre of the image. However, traffic signs are generally localised towards the edges of the image and are smaller. YOLO V2 and SSD models show poor performance on small traffic sign images despite reaching accuracy scores better than or similar

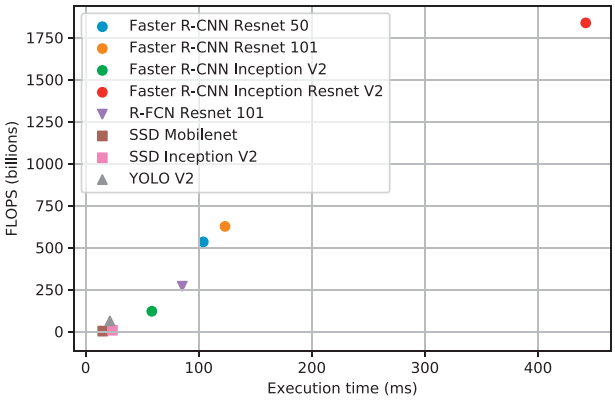


Fig. 4. FLOPS vs. execution time.

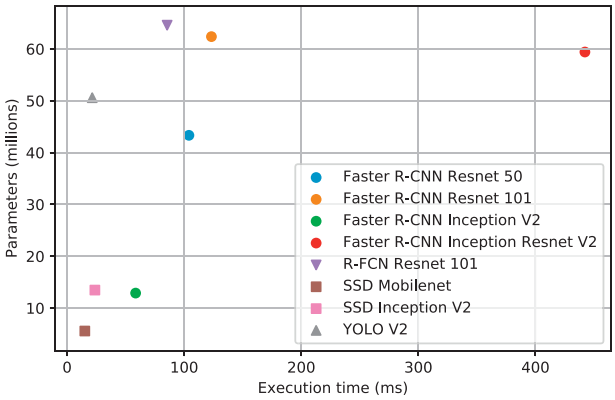


Fig. 5. Parameters vs. execution time.

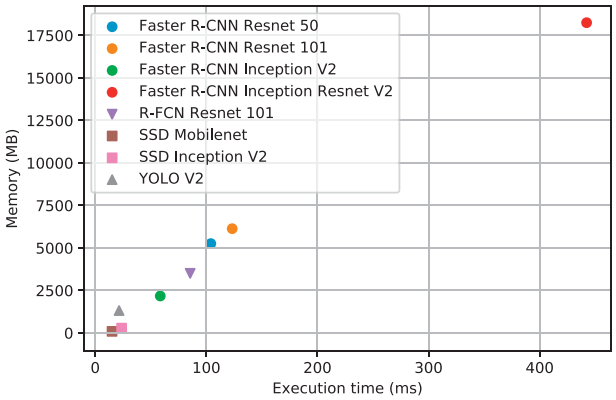


Fig. 6. Memory vs. execution time.

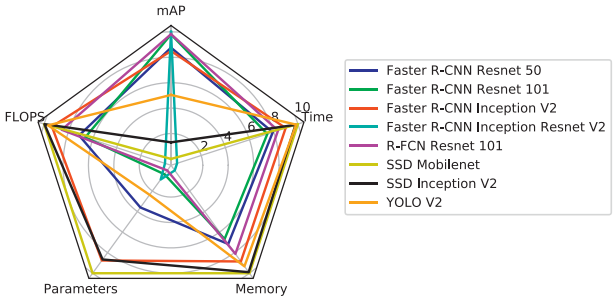


Fig. 7. Analysis overview of the traffic sign detector models.

to Faster R-CNN and R-FCN models on large traffic sign samples. This insight can also be observed in the papers where the models were originally described by their authors [5–8] and in [12]. For instance, the YOLO V2 model, trained on the PASCAL VOC 2012 dataset [8], achieves a lower performance in detecting small objects, such as plants (49.1% AP) and bottles (51.8% AP) in comparison with its performance in detecting other kinds of larger objects, such as bikes (82% AP), airplanes (86.3% AP), and cars (76.5% AP).

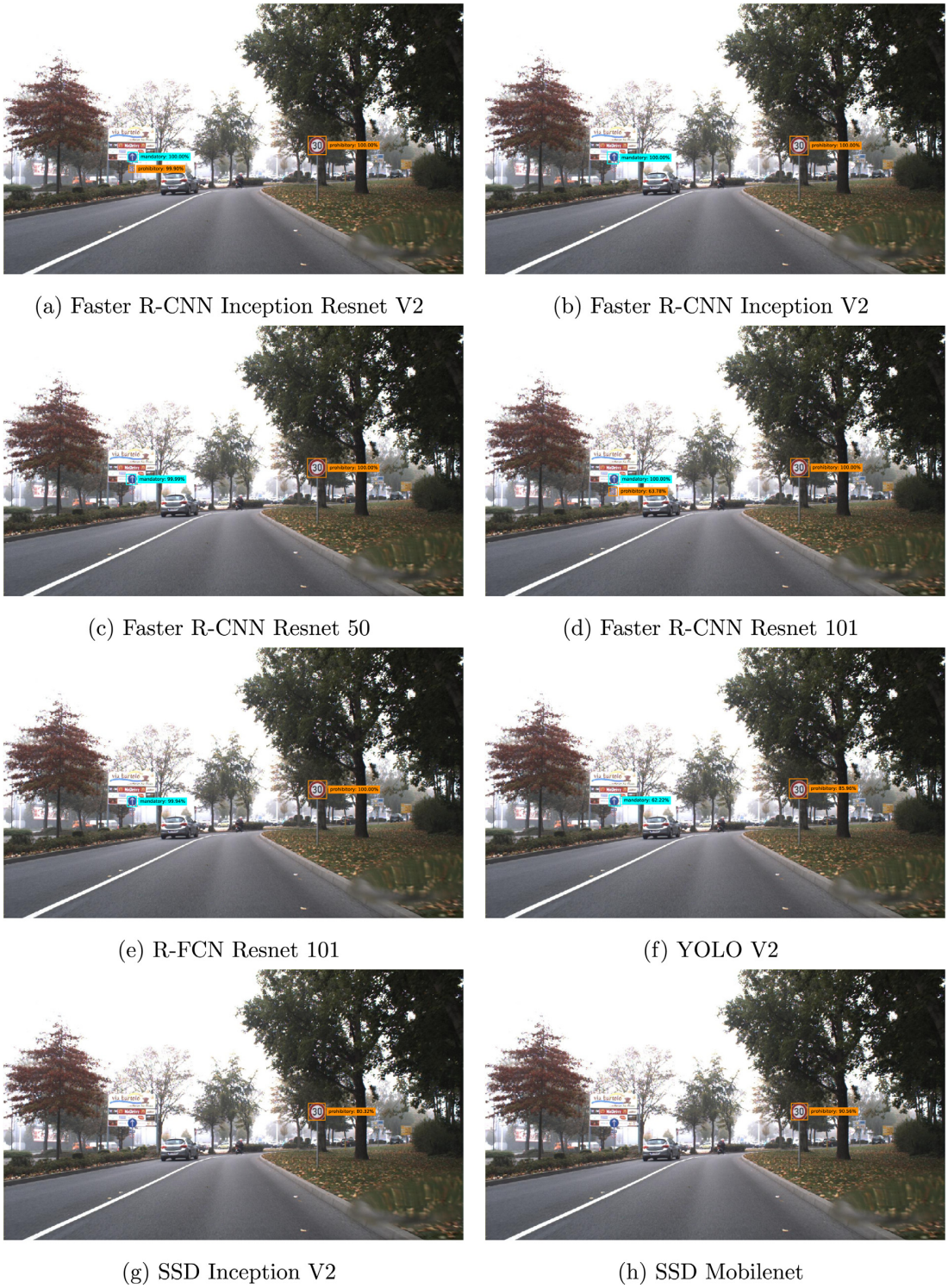
Fig. 4 represents the FLOP count against execution time. The number of FLOPs computed by each model is a platform-independent measure. On one hand, the use of denser blocks in residual networks leads to higher FLOPs and computation time for both Faster R-CNN and R-FCN detectors. On the other hand, SSD Mobilenet is the model with the fewest FLOPs and shortest running time. It should be borne in mind that the FLOP counter may not be linear with respect to actual execution times, due to multiple factors, such as hardware optimisation, and memory I/O. This fact can be observed in the comparison of YOLO V2 and SSD Inception V2 models. The former executes 62.78 billion FLOPs in less time than the latter, which performs 7.59 billion FLOPs. Moreover, the number of parameters that each neural network has to learn (weights and bias) is not directly related with their running time, as shown in Fig. 5. It can be seen that models whose feature extractor is a Resnet 101 contain millions more parameters than detectors with higher (Faster R-CNN Inception Resnet V2) or similar (Faster R-CNN Resnet 50) execution times. YOLO V2 is an analogous case since having approximately 50 million learnable parameters, its computation time is shorter than or nearly equal to that of lightweight models, such as SSD Mobilenet, SSD Inception V2, and Faster R-CNN Inception V2.

Memory consumption is also a critical factor. It helps to make decisions, such as whether a certain model can be trained on a single GPU or whether it is necessary to use a cluster of these computation units, and to decide whether a determined neural network architecture can be deployed in mobile and embedded devices. Fig. 6 plots total memory usage against the running time of the models studied. A high linear correlation exists between execution time and larger and more powerful feature extractors that require much more memory. Again, the models based on residual networks occupy the top positions in terms of memory usage, while SSD Mobilenet and SSD Inception V2 models are the cheapest in that they require 94.70 MB and 284.51 MB, respectively.

Finally, a radar chart is plotted in Fig. 7 whose spokes represent the five measured factors as described above: mAP, execution time, FLOPs, parameters, and memory usage. The minimum value of each measure was considered as the best, except for mAP, where the maximum value was taken as the best. Moreover, for each factor, all values were converted to the range [0,10]. It should be borne in mind that mAP, running time, and memory consumption constitute the most critical factors. Consequently, we observe that the best overall models are R-FCN Resnet 101 and Faster R-CNN Inception V2.

4.3. Traffic sign detections in real-world scenarios

In Figs. 8–10, a side-by-side comparison is presented of the traffic signs detected in images from the GTSDB dataset using the eight detectors evaluated in this paper. The visualised detections have a score value greater than a threshold of 0.5. Three common scenarios are represented in these figures: Firstly, a road scene that contains small, medium-sized, and large traffic signs of different



(a) Faster R-CNN Inception Resnet V2

(b) Faster R-CNN Inception V2

(c) Faster R-CNN Resnet 50

(d) Faster R-CNN Resnet 101

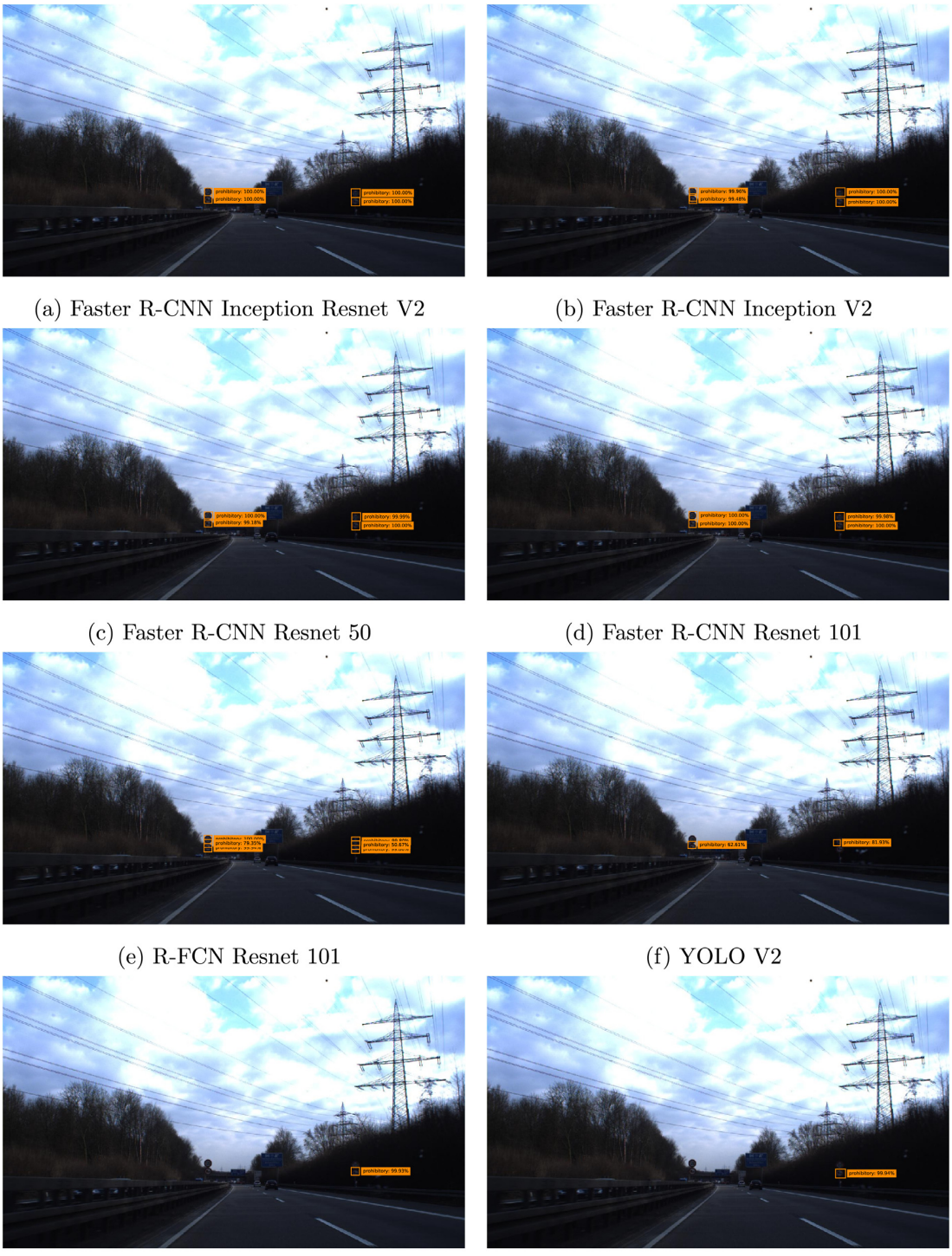
(e) R-FCN Resnet 101

(f) YOLO V2

(g) SSD Inception V2

(h) SSD Mobilenet

Fig. 8. Example detections from 8 different models in a road scene with small, medium-sized and large traffic signs of multiple categories. All detections are correct in examples *a* and *d*. In *b*, *c*, *e* and *f*, the smallest traffic sign is not recognised. Finally, in *g* and *h*, two traffic signs are not localised.



(a) Faster R-CNN Inception Resnet V2

(b) Faster R-CNN Inception V2

(c) Faster R-CNN Resnet 50

(d) Faster R-CNN Resnet 101

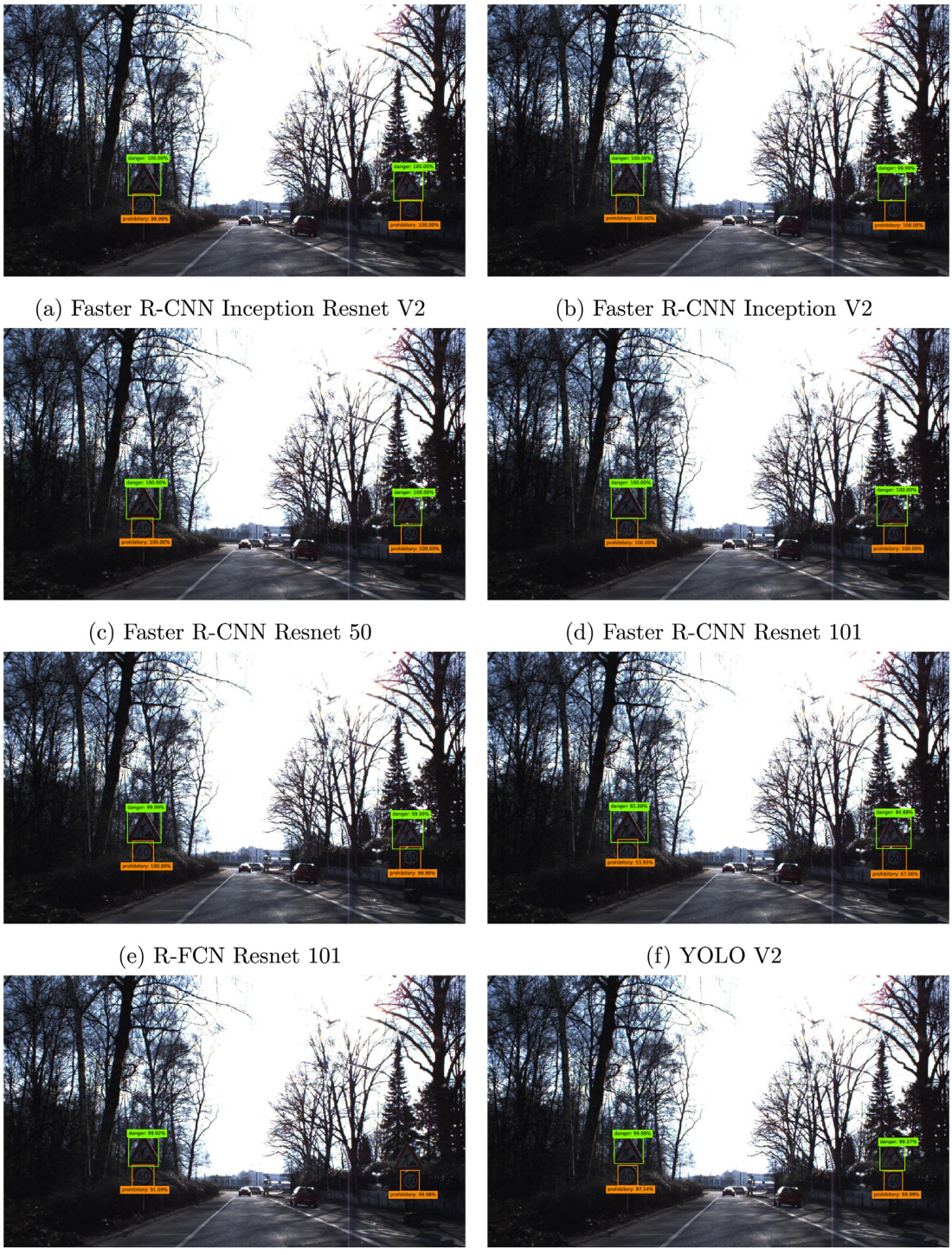
(e) R-FCN Resnet 101

(f) YOLO V2

(g) SSD Inception V2

(h) SSD Mobilenet

Fig. 9. Example detections from 8 different models in a road scene with small traffic signs of the same type grouped on both sides of the road. All detections are correct in examples *a*, *b*, *c* and *d*. In *e*, there are two false positives. Two traffic signs remain undetected in *f*. Finally, in *g* and *h* three traffic signs are not recognised.



(a) Faster R-CNN Inception Resnet V2

(b) Faster R-CNN Inception V2

(c) Faster R-CNN Resnet 50

(d) Faster R-CNN Resnet 101

(e) R-FCN Resnet 101

(f) YOLO V2

(g) SSD Inception V2

(h) SSD Mobilenet

Fig. 10. Example detections from 8 different models in a road scene with large traffic signs of various categories grouped on both sides of the road. Only in *g* is there an undetected traffic sign.

categories (Fig. 8); Secondly, an image where small grouped traffic signs located on both sides of the road can be visualised (Fig. 9); Thirdly, a scene where multiple large traffic signs of various categories are grouped and localised on both sides of the road (Fig. 10). We observe that all of the detectors perform well on large traffic signs. However, YOLO V2 and SSD models are weak at detecting small traffic signs, especially when these signs appear in groups. Additionally, detection scores are generally lower than those of the remaining detectors. Furthermore, YOLO V2 has certain limitations. It imposes strong spatial constraints on bounding box predictions since each grid cell can only have one class. This restricts the set of possible predictions in the case where there are many nearby objects, which is the case represented in Fig. 9, where multiple small traffic signs appear in groups as mentioned above. Other models, such as R-FCN Resnet 101, and Faster R-CNN Inception V2, also present difficulties in detecting signs in this image because they have some false positives localised very near to the real true positives. It is remarkable that only Faster R-CNN Inception Resnet V2 and Faster R-CNN Resnet 101 models are able to detect every traffic sign included in the scene shown in Fig. 8. With scores near to 100%, they even manage to detect a prohibitory traffic sign (the smallest sign) on the left-hand size of the image, which was not annotated in the ground truth.

5. Conclusions and future work

In this paper, a experimental comparison of eight traffic sign detectors based on deep neural networks is presented. We analyse the main aspects of these detectors, such as accuracy, speed, memory consumption, number of floating point operations, and number of learnable parameters within the CNN. All of the models studied in this work were pre-trained on the Microsoft COCO dataset and fine-tuned afterwards with the GTSDB dataset in order to detect and classify traffic sign superclasses based on their shapes and colours: mandatory, prohibitory, and danger.

Accuracy results are evaluated following the mAP quantitative measure from PASCAL VOC 2010. We found that Faster R-CNN Inception Resnet V2 obtains the best mAP (95.77%), while R-FCN Resnet 101 holds the best trade-off between accuracy (95.15%) and execution time (85.45 ms per image). Special mentions are deserved by YOLO V2 and SSD Mobilenet. The former achieves competitive accuracy results (78.83%) and is the second-fastest detector with running times of 21.48 ms per image on average. The latter is the fastest model of all of the detectors and also the least demanding in terms of memory consumption. These key factors make SSD Mobilenet an optimal choice for deployment in mobile and embedded devices. Nevertheless, we observed that SSD models remain very weak at detecting small traffic signs despite the fact that it is critical for any TSDS to perform well at detecting signs in advance so that correct decisions can be made as soon as possible. In general, all of the models present good results at detecting large traffic signs (mAP above 75%). It is also very interesting that only the YOLO V2 and SSD models achieve more than 30 FPS using a NVIDIA Titan Xp, which makes them feasible for real-time traffic sign detection. Another conclusion is that the application of transfer learning to pre-trained models leads to results close to those obtained with the state-of-the-art methods in a specific domain, such as traffic sign detection, where the best results are achieved using a CNN with dilated convolutions on 5 different image scales [23].

It should be borne in mind that the evaluation of the detectors was performed on isolated traffic scene images recorded on various types of roads. Hence, the images are not continuous in time and, consequently, tracking systems could not be used. Such tracking systems could improve the performance of the detectors if the source of the images were made up of consecutive frames extracted from a video.

In future work, we plan to research other neural network architectures that have been proven to work well detecting or classifying general-purpose objects, such as DenseNet [45], and to adapt them to the traffic sign recognition domain. Moreover, advanced embedded platforms, such as NVIDIA Jetson TX2[6] and NVIDIA Drive Px,[7] have recently been released: the detectors presented in this paper should be evaluated using these new platforms in order to reveal valuable insights that help practitioners choose and deploy an appropriate traffic sign detector in the real world.

Acknowledgments

This work has been partially supported by the projects “Hermes-Smart Citizen” and “VICTORY” (both MINECO/FEDER R&D, UE) (Grant nos.: TIN2013-46801-C4-1-R and TIN2017-82113-C2-1-R). We would like to thank NVIDIA for the Titan Xp GPU donated to our research team via the academic GPU seeding program.

References

[1] A. De La Escalera, L.E. Moreno, M.A. Salichs, J.M. Armingol, Road traffic sign detection and classification, IEEE Trans. Ind. Electron. 44 (6) (1997) 848–859, doi:10.1109/41.649946.
[2] Á. Arcos-García, J.A. Álvarez-García, L.M. Soria-Morillo, Deep neural network for traffic sign recognition systems: an analysis of spatial transformers and stochastic optimisation methods, Neural Netw. 99 (2018) 158–165.
[3] L. Zhou, Z. Deng, Lidar and vision-based real-time traffic sign detection and recognition algorithm for intelligent vehicle, in: Proceedings of the 2014 IEEE 17th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2014, pp. 578–583.
[4] Á. Arcos-García, M. Soilán, J.A. Álvarez-García, B. Riveiro, Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems, Expert Syst. Appl. 89 (2017) 286–295.
[5] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Proceedings of Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[6] J. Dai, Y. Li, K. He, J. Sun, R-fcn: object detection via region-based fully convolutional networks, in: Proceedings of Advances in Neural Information Processing Systems, 2016, pp. 379–387.
[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single Shot Multibox Detector, 9905 LNCS, 2016, pp. 21–37, doi:10.1007/978-3-319-46448-0_2. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
[8] J. Redmon, A. Farhadi, Yolo9000: Better, Faster, Stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525, doi:10.1109/CVPR.2017.690.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2009, pp. 248–255.
[10] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338, doi:10.1007/s11263-009-0275-4.
[11] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common Objects in Context, 8693 LNCS, 2014, pp. 740–755, doi:10.1007/978-3-319-10602-1_48. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, K. Murphy, Speed/accuracy trade-offs for modern convolutional object detectors, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3296–3297, doi:10.1109/CVPR.2017.351.
[13] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of traffic signs in real-world images: the German traffic sign detection benchmark, in: Proceedings of the 2013 International Joint Conference on Neural Networks, IJCNN, IEEE, 2013, pp. 1–8.
[14] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[15] M. Liang, M. Yuan, X. Hu, J. Li, H. Liu, Traffic sign detection by roi extraction and histogram features-based recognition, in: Proceedings of the 2013 International Joint Conference on Neural Networks, IJCNN, IEEE, 2013, pp. 1–8.
[16] M. Mathias, R. Timofte, R. Benenson, L. Van Gool, Traffic sign recognitionâ;;how far are we from the solution? in: Proceedings of the 2013 International Joint Conference on Neural Networks, IJCNN, IEEE, 2013, pp. 1–8.

[6] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/
[7] https://www.nvidia.com/en-us/self-driving-cars/drive-px/

[17] G. Wang, G. Ren, Z. Wu, Y. Zhao, L. Jiang, A robust, coarse-to-fine traffic sign detection method, in: Proceedings of the 2013 International Joint Conference on Neural Networks, IJCNN, 2013, pp. 1–5, doi:10.1109/IJCNN.2013.6706812.
[18] D. Zang, J. Zhang, D. Zhang, M. Bao, J. Cheng, K. Tang, Traffic sign detection based on cascaded convolutional neural networks, in: Proceedings of 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD, 2016, pp. 201–206, doi:10.1109/SNPD.2016.7515901.
[19] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[20] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, W. Liu, Traffic sign detection and recognition using fully convolutional network guided proposals, Neurocomputing 214 (2016) 758–766.
[21] C.L. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges, in: Proceedings of European Conference on Computer Vision, Springer, 2014, pp. 391–405.
[22] F. Larsson, M. Felsberg, Using fourier descriptors and spatial models for traffic sign recognition, in: Proceedings of Scandinavian Conference on Image Analysis, Springer, 2011, pp. 238–249.
[23] H.H. Aghdam, E.J. Heravi, D. Puig, A practical approach for detection and classification of traffic signs using convolutional neural networks, Robot. Auton. Syst. 84 (2016) 97–112.
[24] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: ICLR, 2016.
[25] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, OverFeat: Integrated recognition, localization and detection using convolutional networks, in: ICLR, 2014.
[26] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[27] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[28] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916, doi:10.1109/TPAMI.2015.2389824.
[29] R. Girshick, Fast R-CNN, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi:10.1109/ICCV.2015.169.
[30] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in: Proceedings of Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.
[31] A. Mogelmose, M.M. Trivedi, T.B. Moeslund, Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey, IEEE Trans. Intell. Transp. Syst. 13 (4) (2012) 1484–1497.
[32] R. Timofte, K. Zimmermann, L. Van Gool, Multi-view traffic sign detection, recognition, and 3D localisation, Mach. Vis. Appl. 25 (3) (2011) 633–647, doi:10.1007/s00138-011-0391-3.
[33] F. Jurišić, I. Filković, Z. Kalafatić, Multiple-dataset traffic sign classification with onecnn, in: Proceedings of 2015 3rd IAPR Asian Conference on Pattern Recognition, ACPR, IEEE, 2015, pp. 614–618.
[34] A. Youssef, D. Albani, D. Nardi, D.D. Bloisi, Fast traffic sign recognition using color segmentation and deep convolutional networks, in: Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems, Springer, 2016, pp. 205–216.
[35] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, Traffic-sign detection and classification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2110–2118.
[36] N. Qian, On the Momentum Term in Gradient Descent Learning Algorithms, 1999. 10.1016/S0893-6080(98)00116-6
[37] T. Tieleman, G. Hinton, Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude, 2012, (COURSERA: Neural Networks for Machine Learning).
[38] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, Vol. 37 of Proceedings of Machine Learning Research, PMLR, Lille, France, 2015, pp. 448–456.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[40] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of International Conference on Machine Learning, 2015, pp. 448–456.
[41] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning., in: Proceedings of AAAI, 2017, pp. 4278–4284.
[42] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).
[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of International Conference on Learning Representations, ICRL, 2015, pp. 1–14, doi:10.1016/j.infsof.2008.09.005.
[44] M. Lin, Q. Chen, S. Yan, Network in Network, in: ICLR, 2014.
[45] G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017, p. 3.

Álvaro Arcos García received his Computer Science Master's degree from University of Seville (Spain) in 2013 and is currently pursuing his Ph.D. in Computer Science at the University of Seville, Spain. His research interests include deep learning, computer vision, ubiquitous computing and cloud computing.

Juan A. Álvarez García is an Associate Professor at the Department of Languages and Computer Systems at the University of Seville. His research focuses in computer vision, deep learning and human activity recognition. He holds a Ph.D. degree in Software Engineering from the University of Seville.

Luis M. Soria Morillo is a Lecturer and Researcher at the Department of Languages and Computer Systems at the University of Seville. His research focuses in activity recognition, human computer interaction and healthcare solutions. He holds a Ph.D. degree in Software Engineering from the University of Seville.

PARTE III

Observaciones finales

CAPÍTULO 5

CONCLUSIONES Y TRABAJO FUTURO

Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning. - Albert Einstein

5.1. Conclusiones

Esta tesis se centra en el problema de desarrollar un sistema de reconocimiento de señales de tráfico sobre imágenes 2D robusto y en tiempo real, lo cual presenta dificultades en términos de precisión y tiempo de ejecución. Por ejemplo, la aplicación de estos sistemas en vehículos autónomos debe cumplir requisitos estrictos para que la toma de decisiones sea la correcta dado un contexto determinado. Un sistema de reconocimiento de señales de tráfico está compuesto por dos etapas: detección y clasificación. La primera se centra en localizar en imágenes de escenarios de carreteras las señales mientras que la segunda ejecuta una clasificación fina para

identificar qué tipo de señal es.

En el contexto de detección de señales, en este trabajo se investigan ocho modelos de redes neuronales profundas para conocer sus propiedades, entre las que se encuentran la precisión, la velocidad de ejecución, el consumo de memoria, el número de operaciones de punto flotante, el número de parámetros del modelo, y por último, cómo se comportan dichas redes con distintos tamaños de imágenes de entrada. Además, se aplica el concepto de transferencia de aprendizaje entre redes neuronales. La evaluación final muestra que que Faster R-CNN Inception Resnet V2 alcanza el mejor porcentaje de precisión, mientras que R-FCN Resnet 101 obtiene el mejor equilibrio entre precisión y velocidad de ejecución. Por otro lado, el modelo más rápido es SSD Mobilenet al mismo tiempo que es el que menor consumo de memoria tiene, por lo que es la solución ideal para ser desplegada en dispositivos móviles o embebidos, siempre y cuando la precisión no sea el factor más importante, debido a que no tiene un buen rendimiento detectando señales de tráfico pequeñas. Por último destacar que únicamente los modelos YOLO V2 y aquellos basados en la arquitectura SSD se pudieron ejecutar a más de 30 FPS usando una GPU NVIDIA Titan Xp, lo cual los hace factibles para la detección de señales de tráfico en tiempo real.

Con respecto a la clasificación de señales de tráfico, proponemos una red neuronal profunda que contiene capas convolucionales y redes de transformardores espaciales. Las redes de transformadores espaciales permiten realizar operaciones de transformaciones afines sobre las imágenes y los mapas de características, de modo que la red aprende a centrarse exclusivamente en la señal de tráfico, eliminado el fondo, realizando rotaciones, traslaciones, etc. La inclusión de estos elementos en una red convolucional permitió que el modelo superase a todos los métodos publicados anteriormente en la literatura, estableciendo un nuevo récord de precisión del 99.71 % en el German Traffic Sign Recognition Benchmark [(1)] (Figura 5.1). Hasta la fecha de redacción de esta tesis doctoral, los resultados alcanzados por nuestro método no han sido superados. Además, nuestra propuesta no necesitaba aplicar técnicas de

[(1)] `http://benchmark.ini.rub.de/?section=gtsrb&subsection=results`

TEAM	METHOD	TOTAL	SUBSET (All signs)
[156] DeepKnowledge Seville	CNN with 3 Spatial Transformers	99.71%	99.71%
[3] IDSIA	Committee of CNNs	99.46%	99.46%
[155] COSFIRE	Color-blob-based COSFIRE filters for object recogn	98.97%	98.97%
[1] INI-RTCV	Human Performance	98.84%	98.84%
[4] sermanet	Multi-Scale CNNs	98.31%	98.31%
[2] CAOR	Random Forests	96.14%	96.14%
[6] INI-RTCV	LDA on HOG 2	95.68%	95.68%
[5] INI-RTCV	LDA on HOG 1	93.18%	93.18%
[7] INI-RTCV	LDA on HOG 3	92.34%	92.34%

Figura 5.1: Resultados German Traffic Sign Recognition Benchmark.

aumentación de datos manual que se habían realizado en trabajos previos, y requería un menor uso de memoria debido a que el número de parámetros de la red era inferior a otras soluciones.

5.2. Trabajo futuro

Respecto a la detección de señales de tráfico, el trabajo futuro debe enfocarse en investigar nuevas arquitecturas de redes neuronales descritas en la literatura que han funcionado con excelentes resultados detectando objetos generales, adaptándolas al dominio de las señales de tráfico. Además, varias plataformas avanzadas de procesamiento embebido como NVIDIA Jetson TX2 o NVIDIA Drive Px han sido lanzadas al mercado y los detectores propuestos en esta tesis deberían ser evaluados en tales plataformas para obtener información valiosa que ayude a profesionales e investigadores a elegir y desplegar detectores de señales de tráfico para solventar problemas del mundo real.

Por otro lado, en el contexto de la clasificación fina de señales de tráfico, deben estudiarse arquitecturas de redes neuronales que presenten resultados competentes ante señales de distintos países que tengan pictogramas similares, evitando de este modo la necesidad de recolectar conjuntos de imágenes de cada uno de los países donde se desee aplicar el clasificador. Dichos clasificadores deben hacer frente tam-

bién a imágenes creadas con redes generativas antagónicas (Generative Adversarial Networks), las cuales pueden causar efectos negativos en la seguridad vial, poniendo en peligro tanto a conductores como peatones.

APÉNDICE A

CURRICULUM

A.1. Revistas indexadas JCR

1. Título: **Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems**. Autores: **Álvaro Arcos-García, Mario Soilán, Juan A. Álvarez-García, Belén Riveiro**.

   Publicado en: **Expert Systems with Applications**, Elsevier, ISSN: 0957-4174, Fecha de Publicación: Diciembre 2017, Volumen: 89, En Páginas: 286-295, DOI: `https://doi.org/10.1016/j.eswa.2017.07.042`, **Q1 en Computer Science, Artificial Intelligence (20/132). JCR-2017 F.I.: 3.768**.

2. Título: **Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods**. Autores: **Álvaro Arcos-García, Juan A.Álvarez-García, Luis M.Soria-Morillo**.

   Publicado en: **Neural Networks**, Elsevier, ISSN: 0893-6080, Fecha de Publicación: Marzo 2018, Volumen: 99, En Páginas: 158-165, DOI: `https://doi.org/10.1016/j.neunet.2018.01.005`, **Q1 en Computer Science, Artifi-**

cial Intelligence (7/132). JCR-2017 F.I.: 7.197.

3. Título: **Evaluation of deep neural networks for traffic sign detection systems**. Autores: **Álvaro Arcos-García, Juan A.Álvarez-García, Luis M.Soria-Morillo**.

   Publicado en: **Neurocomputing**, Elsevier, ISSN: 0925-2312, Fecha de Publicación: Noviembre 2018, Volumen: 316, En Páginas: 332-344, DOI: `https://doi.org/10.1016/j.neucom.2018.08.009`, **Q1 en Computer Science, Artificial Intelligence (27/132). JCR-2017 F.I.: 3.241**.

A.2. Otras Revistas

4. Título: **Learning in mobility with Context4Learning: developing a context-aware mobile learning application**. Autores: **Marcelo, Carlos, Carmen Yot-Domínguez, Juan Antonio Álvarez-García, Juan Antonio Ortega-Ramírez, and Álvaro Arcos-García**.

   Publicado en: **International Journal of Mobile Learning and Organisation**, ISSN: 1746-7268, Volumen: 10(4), Fecha de Publicación: Agosto 2016, On Pages: 203-222, DOI: `https://doi.org/10.1504/IJMLO.2016.079497`.

A.3. Conferencias Internacionales

5. Título: **Detecting social interactions in working environments through sensing technologies**. Autores: **Juan Antonio Álvarez-García, Álvaro Arcos-García, Stefano Chessa, Luigi Fortunati, Michele Girolami**.

   Publicado en: **Ambient Intelligence-Software and Applications–7th International Symposium on Ambient Intelligence (ISAmI 2016)**, ISSN: 2194-5357, Fecha de Publicación: Mayo 2016, On Pages: 21-29, DOI: `http:`

//dx.doi.org/10.1007/978-3-319-40114-0.

A.4. Conferencias Nacionales

6. Título: **Formarse en la movilidad con Moodle Context. Desarrolo de una aplicación de Mobile Learning sensible al contexto**. Autores: **Marcelo, Carlos, Carmen Yot-Domínguez, Juan Antonio Álvarez-García, and Álvaro Arcos-García**.

   Publicado en: **Adult Learning, Educational Careers and Social Change**, ISBN: 978-84-617-8989-4, Fecha de Publicación: 2017, On Pages: 193-206.

7. Título: **Sistema de reconocimiento de señales de tráfico para una SmartCity**. Autores: **Arcos-García, Á., Soilán, M., Riveiro, B., Álvarez-García, J.A., Fernández, J.Y., Ortega J.A., Arias-Sánchez, P.**.

   Publicado en: **Proceedings of the XVII ARCA Days. Qualitative Systems and its Applications in Diagnose, Robotics, Ambient Intelligence and Smart Cities**, ISBN:978-84-608-5599-6, Fecha de Publicación: Junio 2015, On Pages: 57-62.

8. Título: **Plataforma para gestión de información de ciudadanos de una SmartCity**. Autores: **Jorge Yago Fernández-Rodríguez, Álvaro Arcos García, Juan Antonio Álvarez-García, Jeús Torres, Jesús Arias Fisteus, Víctor Corcoba Magaña, Mario Muñoz Organero, Luis Sánchez Fernández**.

   Publicado en: **Proceedings of the XVII ARCA Days. Qualitative Systems and its Applications in Diagnose, Robotics, Ambient Intelligence and Smart Cities**, ISBN: 978-84-608-5599-6, Fecha de Publicación: Junio 2015, On Pages: 53-56.

A.5. Proyectos I+D+i

Esta tesis doctoral ha sido desarrollada dentro del contexto de los siguientes proyectos de investigación:

- Título: **Healthy and Efficient Routes in Massive Open-Data Based Smart Cities-Citizen**.

  Investigadores principales: **Juan Antonio Álvarez García**. Entidad: **Gobierno de España. Ministerio de Economía y Competitividad**. Periodo: **2014-2017**. Referencia: **TIN2013-46801-C4-1-R**.

- Título: **Vision and Crowdsensing Technology for an Optimal Response in Physical-Security**.

  Investigadores principales: **Juan Antonio Álvarez García, Fernando Enríquez de Salamanca Ros**. Entidad: **Gobierno de España. Ministerio de Economía y Competitividad**. Periodo: **2018-2020**. Referencia: **TIN2017-82113-C2-1-R**.

BIBLIOGRAFÍA

[1] H. H. Aghdam, E. J. Heravi, and D. Puig. A practical approach for detection and classification of traffic signs using convolutional neural networks. *Robotics and autonomous systems*, 84:97–112, 2016.

[2] N. Barnes, G. Loy, and D. Shaw. The regular polygon detector. *Pattern Recognition*, 43(3):592–602, 2010.

[3] N. Barnes, A. Zelinsky, and L. S. Fletcher. Real-time speed sign detection using the radial symmetry detector. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):322–332, 2008.

[4] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012.

[5] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[6] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[7] A. De La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol. Road traffic sign detection and classification. *IEEE transactions on industrial electronics*, 44(6):848–859, 1997.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, 2009.

[9] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3D LIDAR point clouds. In *2011 IEEE International Conference on Robotics and Automation*, 2011.

[10] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[11] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[12] European Union Road Federation. An ERF position paper for maintaining and improving a sustainable and efficient road network. Technical report, European Union, 2015.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[14] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[15] R. Girshick. Fast R-CNN. *arXiv preprint arXiv:1504.08083*, 2015.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[17] A. Gressin, C. Mallet, J. Demantké, and N. David. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS J. Photogramm. Remote Sens.*, 79:240–251, 2013.

[18] A. Gudigar, S. Chokkadi, U. Raghavendra, and U. R. Acharya. Local texture patterns for traffic sign recognition using higher order spectra. *Pattern Recognition Letters*, 94:202 – 210, 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8, 2013.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[23] D.-S. Huang. Systematic theory of neural networks for pattern recognition. *Publishing House of Electronic Industry of China, Beijing*, 201, 1996.

[24] D.-s. Huang. Radial basis probabilistic neural networks: Model and application. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(07):1083–1101, 1999.

[25] D.-S. Huang and J.-X. Du. A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. *IEEE Transactions on Neural Networks*, 19(12):2099–2115, 2008.

[26] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[27] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and Others. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016.

[28] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, and Others. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

[29] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[30] M. Jaderberg Karen Simonyan, and Andrew Zisserman. Spatial transformer networks. *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[31] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, 2009.

[32] J. Jin, K. Fu, and C. Zhang. Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks. *IEEE Trans. Intell. Transp. Syst.*, 15(5):1991–2000, 2014.

[33] F. Jurisic, I. Filkovic, and Z. Kalafatic. Multiple-dataset traffic sign classification with OneCNN. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 614–618, 2015.

[34] S. Kaplan Berkaya, H. Gunduz, O. Ozsen, C. Akinlar, and S. Gunal. On circular traffic sign detection and recognition. *Expert Systems with Applications*, 48:67–75, 2016.

[35] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015*, pages 1–15, 2015.

[36] C. R. Kothari. *Research methodology: Methods and techniques*. New Age International, 2004.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[38] F. Larsson and M. Felsberg. Using Fourier descriptors and spatial models for traffic sign recognition. In *Scandinavian Conference on Image Analysis*, pages 238–249, 2011.

[39] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[40] J. Li, X. Mei, D. Prokhorov, and D. Tao. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE transactions on neural networks and learning systems*, 28(3):690–703, 2017.

[41] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu. Traffic sign detection by ROI extraction and histogram features-based recognition. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8, 2013.

[42] M. Lin, Q. Chen, and S. Yan. Network In Network. *arXiv preprint*, page 10, 2013.

[43] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8693 LNCS, pages 740–755, 2014.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 9905 LNCS, pages 21–37, 2016.

[45] G. Loy and N. Barnes. Fast shape-based road sign detection for a driver assistance system. *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 1:70–75 vol.1, 2004.

[46] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool. Traffic sign recognition—How far are we from the solution? In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8, 2013.

[47] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

[48] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[49] Y. Nesterov. A method of solving a convex programming problem with convergence rate O (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[50] L. Oliveira, U. Nunes, P. Peixoto, M. Silva, and F. Moita. Semantic fusion of laser and vision in pedestrian detection. *Pattern Recognition*, 43(10):3648–3659, 2010.

[51] M. Oquab. stnbhwd, feb 2017.

[52] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[53] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[54] S. Pu, M. Rutzinger, G. Vosselman, and S. O. Elberink. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Remote Sens.*, 66(6):S28–S39, 2011.

[55] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias. Review of mobile mapping and surveying technologies. *Measurement*, 46(7):2127–2145, 2013.

[56] I. Puente, H. González-Jorge, B. Riveiro, and P. Arias. Accuracy verification of the Lynx Mobile Mapper system. *Opt. Laser Technol.*, 45:578–586, 2013.

[57] N. Qian. On the momentum term in gradient descent learning algorithms, 1999.

[58] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *arXiv preprint*, 1612, 2016.

[59] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[60] B. Riveiro, L. Díaz-Vilarino, B. Conde-Carnero, M. Soilán, and P. Arias. Automatic Segmentation and Shape-Based Classification of Retro-Reflective Traffic Signs from Mobile LiDAR Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(1):295–303, 2016.

[61] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano. Traffic sign detection via interest region extraction. *Pattern Recognition*, 48(4):1039–1049, 2015.

[62] D. Scherer, A. Müller, and S. Behnke. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In *Lecture Notes in Computer Science*, pages 92–101, 2010.

[63] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv preprint arXiv*, page 1312.6229, 2013.

[64] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale Convolutional Networks. In *The 2011 International Joint Conference on Neural Networks*, pages 2809–2813, 2011.

[65] W. G. Shadeed, D. I. Abu-Al-Nadi, and M. J. Mismar. Road traffic sign detection in color images. *10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003*, 2:890–893, 2003.

[66] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)*, pages 1–14, 2015.

[67] M. Soilán, B. Riveiro, J. Martínez-Sánchez, and P. Arias. Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *ISPRS J. Photogramm. Remote Sens.*, 114:92–101, 2016.

[68] Spanish Government. BOE.es - Documento BOE-A-2000-1546, jan 2003.

[69] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011.

[70] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.*, 32:323–332, 2012.

[71] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *AAAI*, pages 4278–4284, 2017.

[72] M. Tan, B. Wang, Z. Wu, J. Wang, and G. Pan. Weakly Supervised Metric Learning for Traffic Sign Recognition in a LIDAR-Equipped Vehicle. *IEEE Trans. Intell. Transp. Syst.*, 17(5):1415–1427, 2016.

[73] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5087, 2015.

[74] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[75] R. Timofte and L. Van Gool. Iterative nearest neighbors. *Pattern Recognition*, 48(1):60–72, 2015.

[76] R. Timofte, K. Zimmermann, and L. Van Gool. Multi-view traffic sign detection, recognition, and 3D localisation. *Mach. Vis. Appl.*, 25(3):633–647, 2011.

[77] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[78] United Nations Economic Commission for Europe. Convention on road signs and signals, 1968.

[79] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang. A robust, coarse-to-fine traffic sign detection method. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, 2013.

[80] C. Wen, J. Li, H. Luo, Y. Yu, Z. Cai, H. Wang, and C. Wang. Spatial-Related Traffic Sign Inspection for Inventory Purposes Using Mobile Laser Scanning Data. *IEEE Trans. Intell. Transp. Syst.*, 17(1):27–37, 2016.

[81] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *arXiv preprint arXiv:1705.08292*, 2017.

[82] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[83] A. Youssef, D. Albani, D. Nardi, and D. D. Bloisi. Fast traffic sign recognition using color segmentation and deep convolutional networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 205–216, 2016.

[84] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[85] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang. Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.*, 113:106–123, 2016.

[86] F. Zaklouta, B. Stanciulescu, and O. Hamdoun. Traffic sign classification using K-d trees and Random Forests. In *The 2011 International Joint Conference on Neural Networks*, pages 2151–2155, 2011.

[87] D. Zang, J. Zhang, D. Zhang, M. Bao, J. Cheng, and K. Tang. Traffic sign detection based on cascaded convolutional neural networks. In *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 201–206, 2016.

[88] L. Zhou and Z. Deng. LIDAR and vision-based real-time traffic sign detection and recognition algorithm for intelligent vehicle. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 578–583, 2014.

[89] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu. Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, 214:758–766, 2016.

[90] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2110–2118, 2016.

[91] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405, 2014.