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/© 2017 Elsevier Ltd. All rights reserved.

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 retro-reflective 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 pre-processed 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 retro-reflective 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).

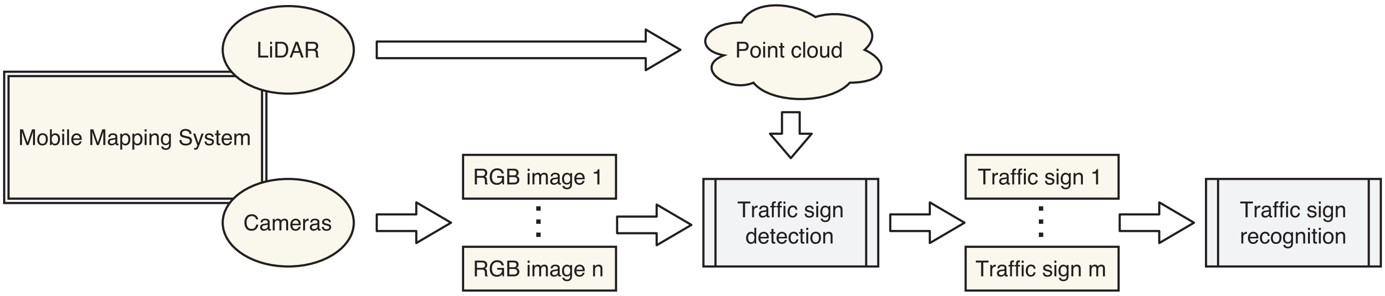

Fig. 1. Proposed methodology. Traffic sign detection by means of LiDAR data processing and traffic sign recognition through a DNN.

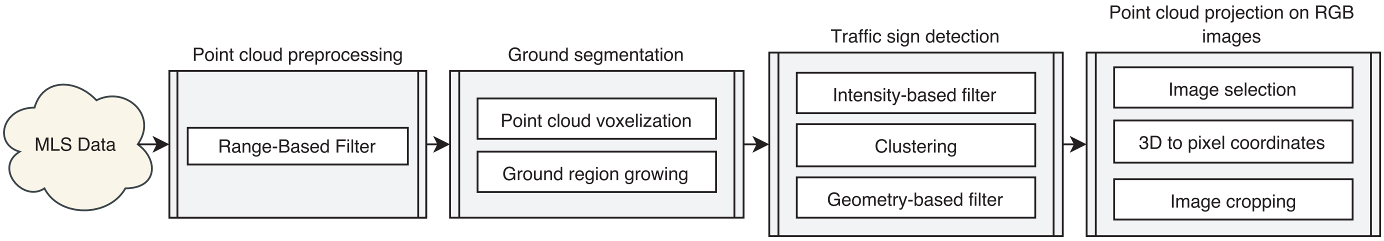

Fig. 2. Point cloud processing. Workflow of the point cloud processing methodology.

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 $V(P) = (x, y, z, \mu_z, \nu_z)$ be the voxelized point cloud of P, and $V(P, id^v) = (x, y, z, \mu_z, \nu_z)$ be the voxel with index id^v, where (x, y, z) is the centroid, and (μ_z, ν_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.

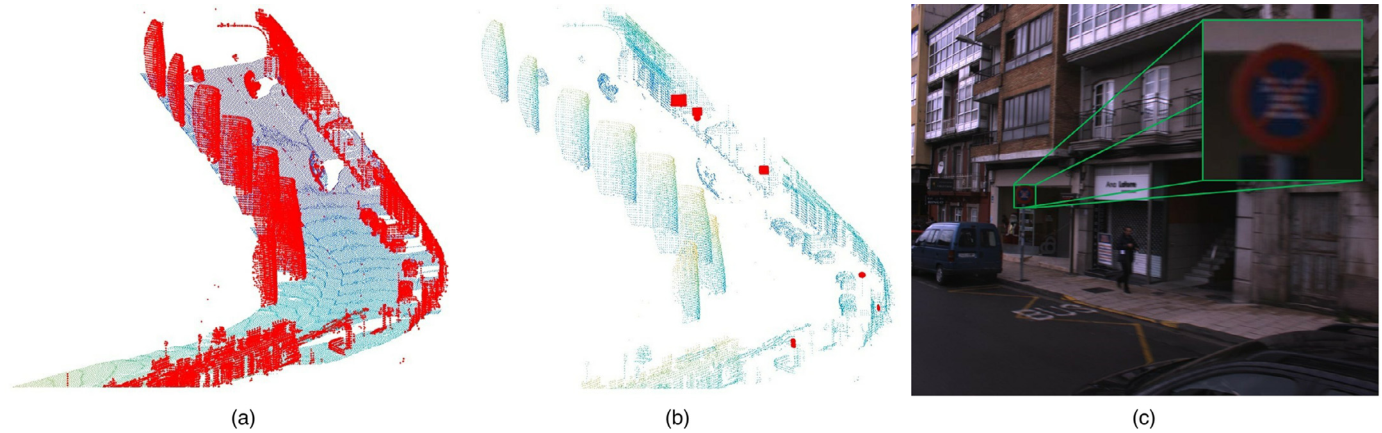

Fig. 3. Traffic sign detection. (a) The ground segment is filtered out from the point cloud. Therefore, only non-ground points (colored in red) are analyzed in the subsequent steps. (b) Both intensity and geometry filters are applied in order to segment traffic sign panels (colored in red). (c) The 3D 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)):

$$p^c_{ih} = (T_{ab} T_{ac})^{-1} p^A_{ih} \tag{6}$$

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

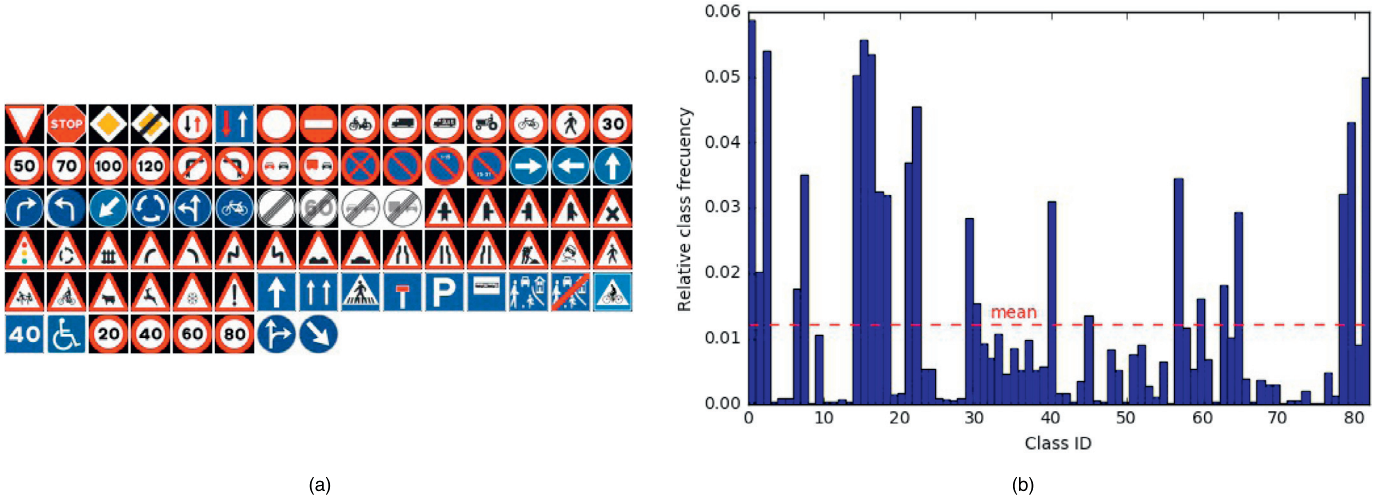

Fig. 4. Mixset dataset. (a) Traffic sign categories. (b) Relative class frequencies.

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

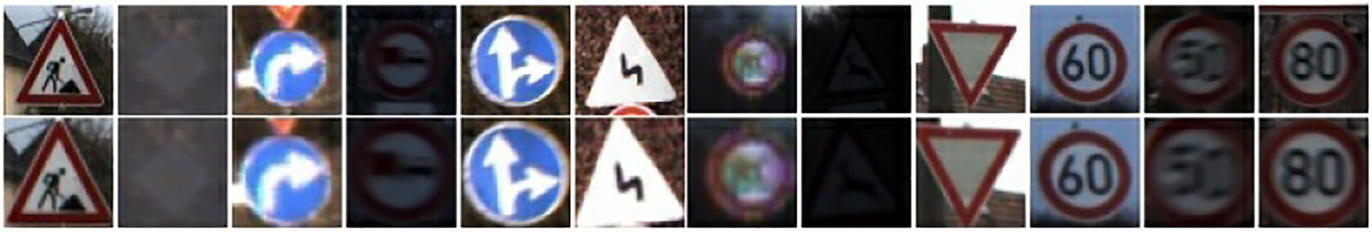

Fig. 5. Spatial transformer network. Input images on the first row and computed output images on the second row.

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.

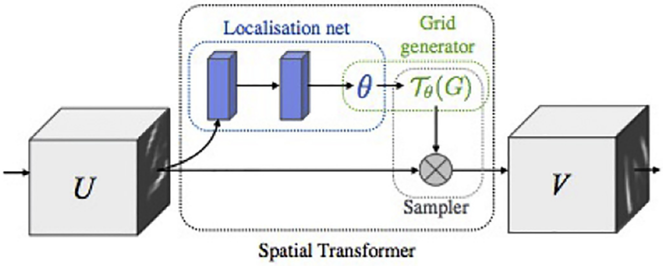

Fig. 6. Spatial transformer network components (Jaderberg et al., 2015).

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

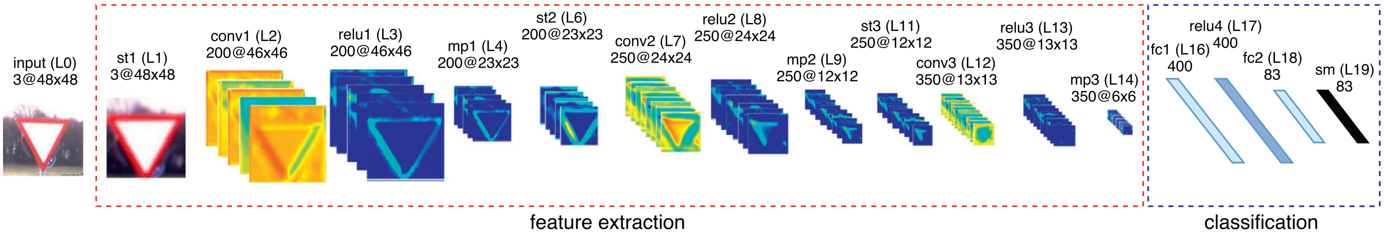

Fig. 7. DNN for traffic sign recognition proposed. Local contrast normalization layers have been omitted in the figure above to simplify its visualization as well as localization networks of spatial transformers. The *st* layers refer to spatial transformer networks, *conv* to convolutional layers, *mp* 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)

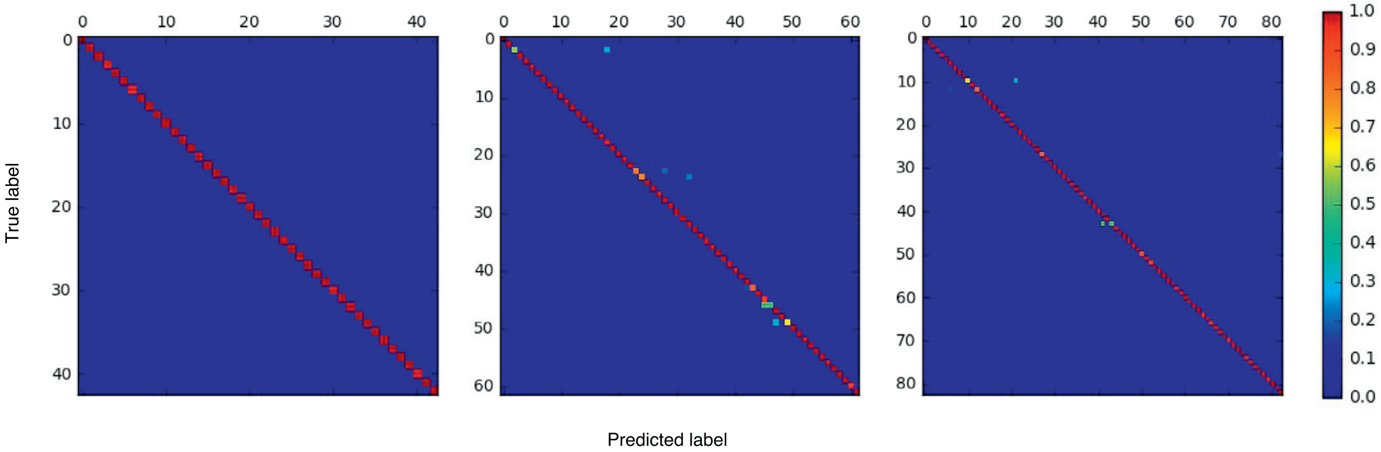

Fig. 8. Confusion matrices. GTSRB on the left, BTSC in the middle and Mixset on the right.

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

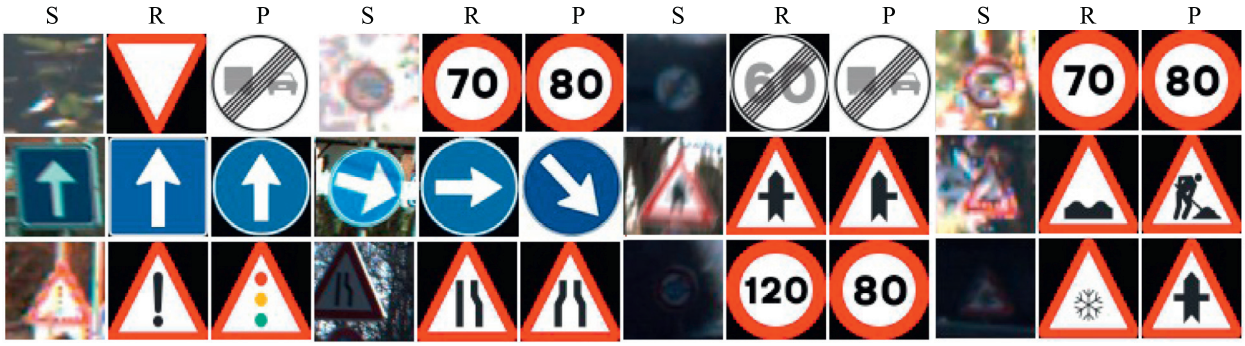

Fig. 9. Misclassified samples. Some misclassified samples of the Mixset model trained. As may be seen, the main reason behind them are occlusions and blurred pictographs, being their recognition even hard for the human visual system. Columns labeled with *S* refer to sample, *R* to real traffic sign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