Deep Extreme Learning Machines with Auto Encoder for Speed Limit Signs Recognition

Óscar Mata-Carballeira[1], Inés del Campo[2], Victoria Martínez[3] and Javier Echanobe[4]

***Abstract*— This work presents a Deep Extreme Learning Machine with Auto Encoder scheme for Speed Limit Signs Recognition in the field of Advanced Driving Assistance Systems, where traffic sign recognition from video imaging plays an important role specially to provide vehicles with automated speed limits enforcement. Current solutions adopted by car manufacturers do not provide robust enough recognition behaviors when the image quality, the lighting conditions or the clearance of the traffic sign are compromised. These conditions result in misinterpreting of the speed limits, showing wrong on-screen advices which might confuse the driver, causing dangerous situations.**

In this work, the full chain of operations is studied. The proposed scheme is trained and tested with the German Traffic Sign Recognition Benchmark (GTSRB) database, achieving recognition times as short as 0.62 ms per sample, reaching with this timing real-time operation, and an accuracy of up to 92% with a simpler structure than other techniques currently used, such as Convolutional Neural Networks (CNNs).

I. Introduction

In recent years, due to the massive improvements in boarded electronics in terms of miniaturization and power consumption, as well as cost reductions, Driving Assistance Systems (DAS) both for amenity (e.g. cruise control, adaptive suspension, etc.) and for safety (anti-lock braking system (ABS), electronic brake assistance system (EBAS), electronic stability program (ESP), etc.) have been gradually implemented in cars of all ranges, with the previously quoted DAS even becoming mandatory over the years for all new-production cars.

Although current cars are fitted with DAS, particularly for safety due to government regulations, many improvements in this field can be still performed, specially in the field of Advanced-DAS, such as Adaptive Cruise Control (ACC), which uses radar detection to keep an adequate safety distance with the preceding vehicle, braking and throttling to accommodate the speed, and eventually, stopping the vehicle in case of a more-than-likely collision [1].

Regarding the enhancement of the traffic signs' enforcement, many mid-rangers are being equipped with automated front-video based traffic sign recognition systems, initially showing on-screen warnings, and, in the latest applications, setting automatically speed limits as the setpoint of the adaptive cruise control.

These current commercial approaches show, nevertheless, many problems on recognizing traffic signs under certain conditions of lighting, missing digits and displaying unstable behaviors [2]. These failures result in misinterpreting speed limits, with the high risk it implies for the driver. In the context of highly assisted cars, drivers are likely to devolve the task of reading the speed limit signs in the vehicle systems in the same manner it happens with many other driving assistance systems, such as ABS or ESP [3]. This problem, known as driver's underload, might transform a safety feature in another source of risk [4].

Since the nature of the human driver is well known [4], in the context of an evolving paradigm of traffic, with the short-term horizon of semi-autonomous cars sharing roadways with conventional automobiles, the point of improving the recognition of speed limit signs is crucial to increase the road safety (since over-speed is one of of the main causes of traffic accidents), and consequently, diminishing the number of traffic-related fatalities.

In this context, the development of a more robust solution for recognizing speed limits capable of coping with the variability of the real world, with scale variations, partially obstructed signs, bad lighting and motion blur is needed. For this purpose, Extreme Learning Machine-Auto Encoders have shown their suitability on image reconstruction and re-interpreting in [5] for handwritten number recognition, achieving extreme levels of accuracy. Another advantage of this approach over other alternatives is the simplicity of its formulation.

The main contribution of this work is a Speed Limit Recognition system based on Multi-Layer Extreme Learning Machines (ML-ELM), a type of Deep Extreme Learning Machine consisting of stacks of ELMs on its Auto Encoder variety as its hidden layers. ML-ELMs are simpler both in mathematical complexity and in the use of a single type of hidden layer instead of the grand variety of types CNNs use. However, even though ML-ELMs are simpler solutions than CNNs, acceptable recognition results are obtained.

The remainder of this document is organized as follows: Means of detecting and identifying traffic signs are dealt with

*This work was supported in part by the Spanish Ministry of Economy, Industry and Competitiveness TEC2016-77618-R (AEI/FEDER, UE), and by the Basque Country University UPV/EHU under Grant PPG17/20.

[1]Óscar Mata-Carballeira is a Researcher at the Faculty of Science and Technology, Electricity and Electronics Department, University of the Basque Country (UPV/EHU), 48940 Leioa, Spain oscar.mata@ehu.es

[2]Inés del Campo is a Senior Lecturer at the Faculty of Science and Technology, Electricity and Electronics Department, University of the Basque Country (UPV/EHU), 48940 Leioa, Spain ines.delcampo@ehu.es

[3]Victoria Martínez is a Senior Lecturer at the Faculty of Science and Technology, Electricity and Electronics Department, University of the Basque Country (UPV/EHU), 48940 Leioa, Spain victoria.martinez@ehu.es

[4]Javier Echanobe is a Senior Lecturer at the Faculty of Science and Technology, Electricity and Electronics Department, University of the Basque Country (UPV/EHU), 48940 Leioa, Spain franciscojavier.echanove@ehu.es

along Section II, while the solutions adopted to implement this method are described in Section III. Section IV compiles the results while the conclusions are shown in Section V.

II. Related Works

As indicated in [6], a typical Traffic Sign Recognition integrates three, well-differentiated modules:

A) Image pre-processing and extraction:
- Edge detection.
- Sign detection.
- Sign extraction.

B) Feature extraction.

C) Classification.

In the field of image pre-processing and extraction, the first part is edge detection. Several algorithms can be used on this topic [7]. The first one is Sobel Operator. This approach uses a couple of 3×3 convolution kernels that are to be moved through the pixel grid of the subject image [8]. These kernels designed to respond maximally to the maximum color gradients, which occur on edges. Other similar approaches are the Prewitt's Operator, which is computed like Sobel's with the only difference that the values of the 3×3 convolution kernels differ; the Roberts Cross Operator, similar to Sobel's and Prewitt's but with 2×2 kernels instead, being more sensitive to $45°$ edges [9]; Laplacian of Gaussian is implemented similar to the preceding algorithms, but it requires a Gaussian smoothing operation to be performed on the image to make it more robust to noise [10]; finally Canny edge detectors [11] have demonstrated their suitability to optimally detect boundaries with its superior signal-noise ratio, performing well even in detecting edges on intensively textured images but with the main drawback of being the most complex algorithm and the most computationally intensive solution.

The next task to perform is sign detection or segmentation. This stage extracts the bitmaps corresponding with the Speed Limit Signs from the remainder of the image. This task is a well known research topic. In [12] two paradigms are distinguished: color-based and shape-based methods. On the one hand, the most remarkable shape-based algorithm suitable for speed limit sign detection is Circle Hough transform, which, after performing an edge detection operation, is able to identify circular patterns by intersecting fixed radii circles along the detected edges [13]. Another method is Distance Transform (DT), in which the position of the corners is calculated first and, after that, the distance of each pixel to the corners is computed to form the DT-feature vector [14]. An additional focus is Histograms of Oriented Gradients (HOGs) [15], which, dividing the image in finite-size cells, integrates their gradients over multiple cells.

On the other hand, color-based detectors are focused on bounding a Region of Interest (ROI) from the segmentation of a color image, and then ROI is delimited, a processing stage extracts the Sign. These detectors are, however, susceptible to day-night conditions, lighting and weather changes since they corrupt the color information [16].

In the field of feature extraction, Haar wavelet features can be used. Haar features are the difference of the sum of pixels inside rectangular areas that can be slided and scaled within the original image [17]. Thus, these features are placed in patterns matching with the locations where singular points of each traffic sign are expected to be found. This approach has the main disadvantage of requiring the definition of a feature map for each signal and, despite it brings very high recognition rates under ideal conditions, it is sensitive to orientation, since in works such as [18], rotation of the Haar wavelet features is not contemplated. Scale-invariant feature transforms (SIFTs) and Gabor features are interesting too, while HOGs can be used also for this application. Another approach is that provided by nuclear-and L2,1-norm regularized 2D neighborhood preserving projection (2DNPP) methods for extracting representative 2D image features [19].

Several approaches exist in the field of speed limit sign classification, but many of them make use of some kind of character segmentation method [20], [21], which first recognizes if the sample is a speed limit sign, and in this case, it applies a threshold binary filtering in order to perform the character segmentation and recognize each digit individually. This solution has the main disadvantage of having two intermediate steps, which affect negatively its robustness and dramatically increase computational costs.

In the field of Machine Learning techniques, Artificial Neural Networks, specially Single-Layer Feedforward Networks (SLFNs) have shown their good performance in many application fields, solving both problems of regression and classification. Although conventional approaches, with tuning algorithms such as Backpropagation, perform well in most of the occasions, it is well known that they have a big disadvantage: since their tuning depends on a gradient-based calculus problem, they are very sensitive to local minima, resulting on conditions of non-convergence or even instability issues [22]. In addition, they have the drawback of converging very slowly in a wide variety of cases and they show poor performance for traffic sign recognition.

Another focus is performing no digit segmentation, considering the extracted Sign as a block Deep Neural Networks show adequate performances. With this consideration, the most important approach that is being used currently are Convolutional Neural Networks (CNNs) [23] which agglutinate the Feature Extraction and Classification stages. However, although CNNs have shown great accuracy rates, they suffer from many problems, such as they still rely on Backpropagation gradient-descent methods. In addition, they use a set of different types of layers to extract different kinds of features, and each single architecture must be selected very carefully. Due to these characteristics they also present a very high computational cost.

Extreme Learning Machines are formulated with the aim of solving this problem [24]. Compared with the backpropagation training paradigm, they provide faster training times due to the avoidance of gradient minimization problems, solving a System of Linear Equations instead.

Sharing the approach of ELMs, ELM-Auto Encoders are designed [5] to extract features by means of a properly trained SLFN, avoiding calculations such as gradients, projections, etc. And finally, Multi-Layer ELMs (ML-ELM) are formulated also in [5] as an efficient manner of integrating the feature extraction proficiency of ELM-AEs with classification and regression abilities, providing in one single step the applications that Speed Limit Signs recognition requires: feature extraction and sample classification. They have also the main advantage, compared with CNNs, that all features are extracted by layers computed by means of ELM-AEs, this is, a single topology for every layer, resulting on a simpler algorithm.

III. Block Diagram of the Proposed Method / Methodology

As shown in Figure 1, this proposed Speed Limit Sign Recognition method comprises three stages: 1) the speed limit sign image extraction module by Sobel Edge Detection and Circle Hough Transform; 2) the contrast normalization stage and 3) a multi-layer ELM to get the signals classified correctly.

A. Dataset of Choice

For the development and testing of this work, several datasets containing images of Traffic Signs are used. These datasets have been collected during the last years from a wide variety of countries, mainly from Europe, North America and Eastern Asia. The samples they contain also include Speed Limit Signs.

However, although there is a full range of dataset options to choose, the experiments have been performed on the German Traffic Sign Recognition Benchmark (GTSRB) dataset [25] since it is widely accepted and uses in many other works in this area. In addition, this dataset is specially suitable for performing training and testing operations on a Speed Limit Recognition System, since it compiles a family of 43 different types of traffic signs under a variety of conditions of lighting, perspective, integrity, image resolution, etc.

Thus, GTSRB comprises 39209 images for training, and 12630 images for validation purposes, all of which are composed of a 24-bit RGB map of sizes ranging from 15 $\times$ 15 up to 250 $\times$ 250 pixels.

Due to the approach taken to perform this work, the only images selected will be those containing the speed limit signs. These signs and the values they display in open-traffic roads are regulated and normalized in each country and their appearances are kept very similar throughout Europe.

Since this work is oriented to public roads, the signs intended to be identified are of the classes displayed in Figure 2.

Signals containing values such as 5, 10 or 15 km/h are neither contained in this dataset nor considered for the system since they are not displayed in open-road conditions (they are typical from parking lots).

In order to provide a uniform input for the Speed Limit recognition solution, all images will be re-scaled to 28$\times$28 matrices, resulting in 784 elements per sample. They will

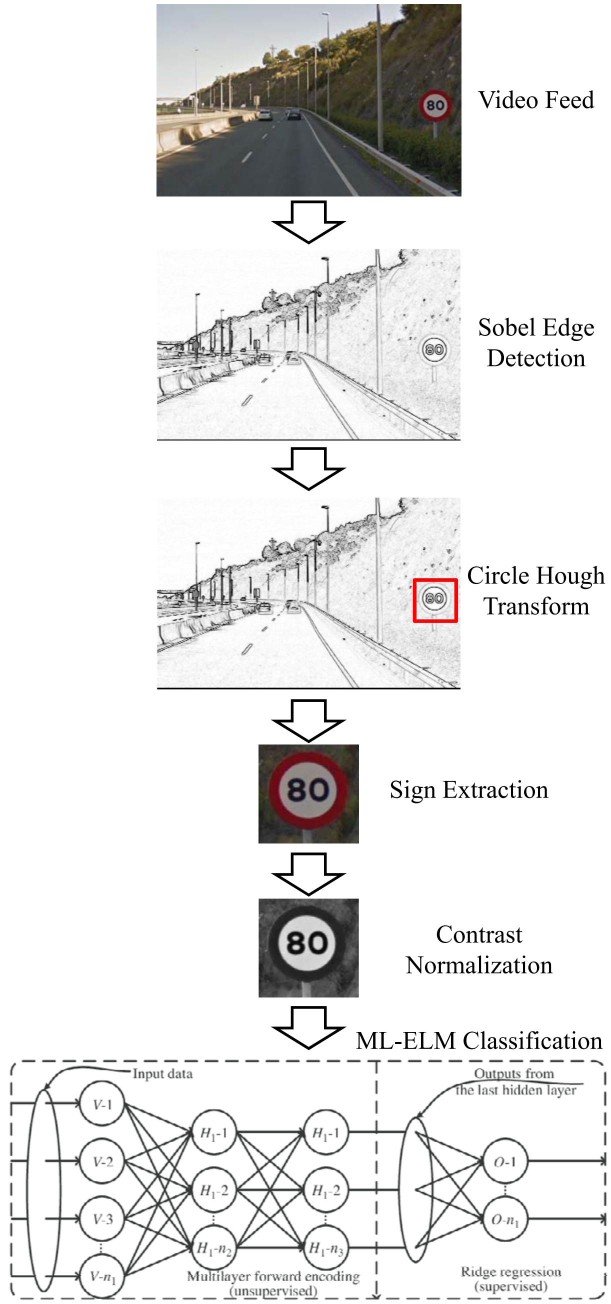

Fig. 1. Block diagram of the proposed solution.

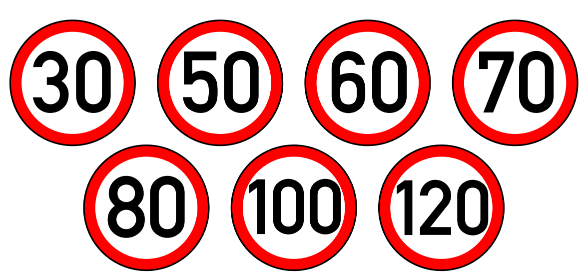

Fig. 2. Normalized European traffic signs (German-Austrian Variant).

also be transformed to gray-scale and contrast normalized. Additionally, since not all training classes contain the same number of samples, their contents are trimmed to have the same number of training experiments for each class, improving the accuracy of the training. By trimming the number of training samples to balance the elements for each class, the training dataset is reduced to 9870 elements (7 classes of 1410 elements each).

B. Traffic Sign Extraction

Concerning the first point, many operations must be performed on the *raw* image, firstly to extract the area of the front-video frame containing the signal we want to identify. For that purpose, several approaches might be carried out such as the Circle Hough Transform. This technique is specifically intended for circular pattern detection on digital imaging since Speed Limit signs are circular in many countries. It is directly derived from the Hough Transform, an algorithm that outperforms approaches such as edge detection, in which, due to imperfections in the pixel structure as well as perspective issues (i.e, circles might appear as ellipses).

Thus, due to the previously pointed out issues, Circle Hough Transform is more robust provided that instead of identifying the round patterns themselves and given a fixed, expected radius or range of radii, it assigns a *score* for each circle candidate. The higher the score, the more likely the candidate is a circle.

The process of finding circular patterns in an image using Circle Hough Transform is subsequently described and graphically represented in Figure 3 [26]:

- Edge detection by Sobel operations [27]
- For each edge point:
  - Draw a circle of the expected radius with center on each edge point.
  - Increment in an accumulator all the coordinates where the drawn circle goes through the detected edges (that will be the score).
  - Find maxima in the accumulator.
- Map the coordinates of the found maxima.

Apart from the circle extraction, which is one of the crucial points of a Traffic Sign Recognition System, the correction of the illumination and contrast levels plays an important role too. There exist several approaches to solve that issue, such as contrast normalization, either in RGB or grayscale images. These operations are very simple, and for gray-scale imaging are based on re-scaling all the brightness values considering the minimum of the original image as absolute black, and the maximum as absolute white.

C. Multi-Layer Extreme Learning Machines

It is well known that the performance of a Machine Learning algorithm depends basically on the concrete features of the input data, so, a manner of extracting those features is crucial to achieve the maximum development in that enterprise. Traditionally, manual engineering of the

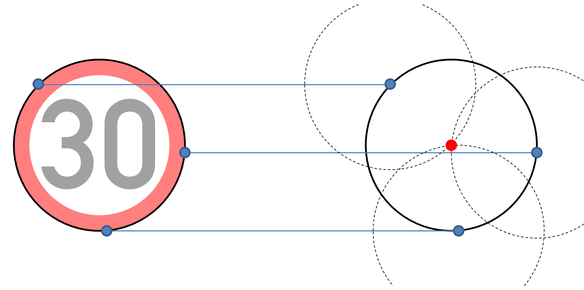

Fig. 3. Graphical interpretation of the Circle Hough Transform.

features has been carried out to extract them from a given dataset, with the corresponding drawbacks it implies, needing knowledge in the area to generate the appropriate features.

In that context, algorithms known as Auto Encoders, can perform the task of generating features automatically, being useful for training multiple-layer neural networks, known as Deep Networks.

Deep Networks outperform Single-Layer Feedforward Networks (SLFNs) and traditional multiple-layer neural networks, but they still show slow learning speeds. Consequently, as a solution to slow learning speeds, Extreme Learning Machines (ELMs) are developed [24] as SLFNs with fast learning rates and improved generalization capabilities, and with the approach shown in [5], where an ELM is used to develop an Auto Encoder (AE), coining the term ELM-AE, Multi-Layer ELMs (ML-ELMs) are deployed as stacks of several ELM-AE, solving the problem of deep networks' slow training rates.

1) Formulation of ELM: The method of ELM proposed [24] assumes that the weights and biases of the hidden nodes of a SLFN are randomly generated. Thus, with L being the number of hidden nodes, the output of an ELM is calculated as follows:

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\beta \tag{1}$$

with $\beta = [\beta_1, \ldots, \beta_L]^T$ the matrix of weights from the hidden layer to the output nodes; $\mathbf{h}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_L(\mathbf{x})]$ are the hidden node outputs (with random features) for the input $\mathbf{x}$; and given a number N of training samples, we define a set of input and training arrays $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_i]$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_i]$ with $i = 1, \ldots, N$. Consequently, ELM is the solution of the following problem:

$$\mathbf{H}\beta = \mathbf{T} \tag{2}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$ are the target labels and $\mathbf{H} = [\mathbf{h}^T(\mathbf{x}_1), \ldots, \mathbf{h}^T(\mathbf{x}_N)]^T$. Thus, since equation 2 is a System of Linear Equations, it can be solved in the subsequent manner:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$

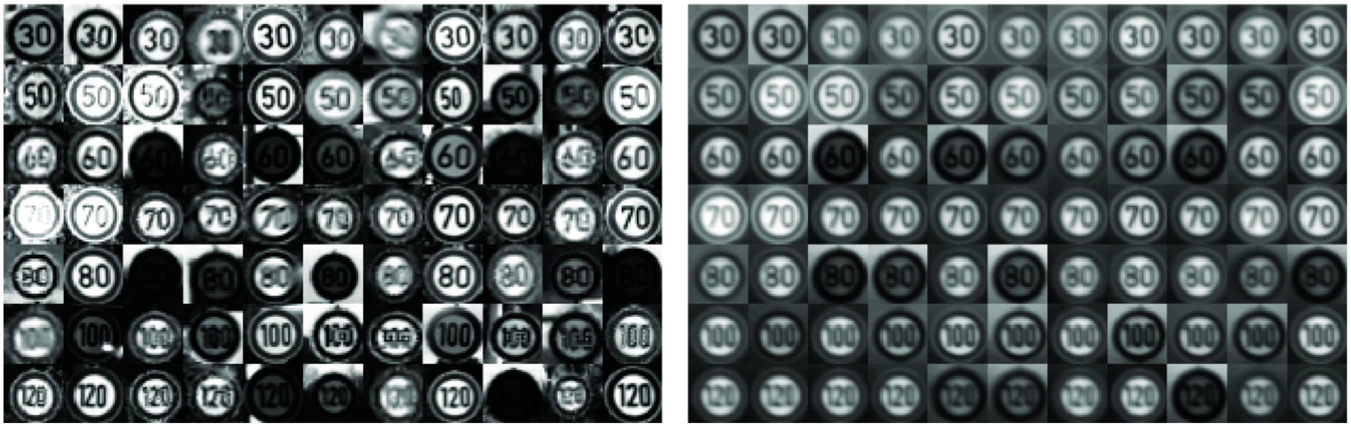

Fig. 4. Samples of different characteristics of lighting, blur and integrity (left) are input to an ELM-AE, resulting in generalized, homogeneous and corrected images for each sample (right).

Nevertheless, depending on the nature of the data, in order to obtain better generalization capabilities and higher robustness, a regularization factor, also known as ridge parameter C, may be added:

$$\beta = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{T} \tag{4}$$

2) ELM as an Auto Encoder (ELM-AE): As shown in [5], there exist three different meaningful input feature representations:

- *Compressed representation*: Represents features in a lower dimensional feature space.
- *Sparse representation*: Represents features in a higher dimensional feature space.
- *Equal-dimension representation*: Represents features in an equal-dimensional feature space.

To train a single ELM-AE, the input data $\mathbf{X}$ is also used as the training set, verifying $\mathbf{T} = \mathbf{X}$. Inserting this equity in Equation 2, we obtain:

$$\mathbf{H}\beta = \mathbf{X} \tag{5}$$

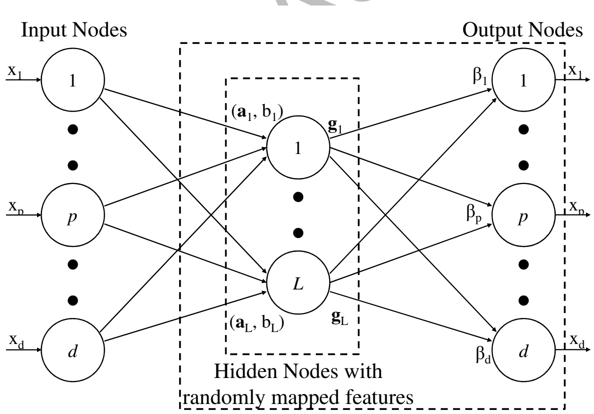

Fig. 5. ELM-AEs are solved in the same way as conventional ELMs. The only exception is that the targets considered for training coincide with the input data. The number of hidden nodes L can be equal, greater or less than the number of input/output nodes.

Concerning the random biases and weights needed to calculate $\mathbf{H}$, although it is not strictly necessary, they should be orthogonalized after they are generated since they tend to provide better unsupervised learning performances [5].

In the field of this work, the advantage ELM-AE brings is that eliminates the deviations shown by each sample returning an intermediate, generalized representation. The behavior of an ELM-AE is displayed on Figure 4. In this figure, a set of 70 speed limit signs (10 samples from 7 different classes) is chosen for both training and testing of an ELM-AE. As it can be observed, with the original preprocessed samples on the left side of the image, and the output of the trained ELM-AE on the right side, a new representation is obtained.

The effect of applying this technique is the removal of blurs, distortions, overexposure of the frame and noisy artifacts, as well as typographic mistakes such as incomplete digits, displacement of the baselines of the digits and alignment errors. It also corrects the orientation of the images, performing a rotation.

In accordance with the behavior displayed on Figure 4, ELM-AEs show their suitability for image processing due to their feature extraction capabilities.

3) Multi-Layer Extreme Learning Machine (ML-ELM): Multi-Layer Extreme Learning Machines may be interpreted as stacks of ELM-AEs. Hence, for each layer we want to add to a ML-ELM, an ELM-AE must be solved. However, differing from the contents displayed in the previous section, since the only information we are interested on is that contained in the hidden nodes, the random orthogonal parameters are discarded in the final stack as they are found implicitly in β.

In addition, considering that ML-ELMs are based on ELM, they do not need to be tuned finely. Thus, applying the postulates of ELM-AE, each ML-ELM hidden layer weights are calculated by finding a β for a ELM-AE assigned to each layer. Hence, given a number of nodes L^j for a set of nodes

TABLE I

AVERAGE RECOGNITION RATES, TEST AND TRAINING TIMES AND STANDARD DEVIATIONS.

Topology	Ridge Params.	Recog. Rate	σ	Test Time	σ	Training Time	σ
784-2050-3200-7000-7	8E-3 180 1500 1E5	90.86%	0.25%	2.59 s	0.10 s	70.55 s	0.82 s
784-2050-3200-7000-7	5E-3 560 29500 1E5	90.95%	0.25%	2.55 s	0.07 s	69.96 s	0.84 s
784-2950-3200-7500-7	8E-3 180 1500 1E5	90.96%	0.35%	2.99 s	0.10 s	83.85 s	0.63 s
784-2950-3200-7500-7	5E-3 560 29500 1E5	91.24%	0.24%	2.96 s	0.10 s	82.87 s	0.59 s
784-2050-3200-15000-7	1E-2 560 29500 1E5	91.65%	0.35%	4.56 s	0.12 s	167.79 s	1.28 s

of the hidden layer $\mathbf{H}^i$, we obtain the set of hidden layer weights β^i as:

$$\beta^i = g(\mathbf{H}^{i-1})^\dagger \mathbf{H}^{i-1} \tag{6}$$

with $g(\mathbf{H}^{i-1})^\dagger$ being the generalized inverse of the array of hidden nodes with randomly mapped features applied to an activation function $g(x)$ according with Figure 5. Assuming β^i as the parameters of the i-th hidden layer, the values of each hidden layer are calculated as follows:

$$\mathbf{H}^i = \left(\beta^i\right)^T \mathbf{H}^{i-1} \tag{7}$$

where $\mathbf{H}^i$ is the vector of nodes of the i-th hidden layer. We will assume that $\mathbf{H}^0 = \mathbf{X}$, the β_{out} of the output layer will be computed using regularized least squares:

$$\beta_{out} = \left(\mathbf{H}^k\right)^\dagger \mathbf{T} \tag{8}$$

with k the number of hidden layers, resulting $\mathbf{H}^k$ the last hidden layer.

IV. RESULTS

The experiments have been run on an Intel Xeon E5-2630 v4 CPU PC with 32 GB of RAM and they are performed in three stages: First, the Circle Hough Transform is checked. Apart, the training reduced dataset is used to infer the parameters of the ML-ELM, training the classifier, and after that, the test dataset is input in the trained classifier and the tags of the testing dataset and the tags inferred by the classifier on the input data are compared, calculating the recognition rates.

The Sobel edge detection and Circle Hough Transform MATLAB implementations are tested over several front video images, resulting in successful extraction of circular traffic signs, among which, speed limit signs are found. For 640×480 front images and a radius of 15 pixels, the signal extraction time is 12$\pm$0.87 ms.

After these checks have been carried out, several trials on different ML-ELM topologies are performed since layers deeper than the first one do not show results as clear as those displayed in Figure 4. Thus, contents of the deeper layers are not directly interpretable considering that they contain transformed features of the preceding layer.

The topologies regarding both the number of nodes for each hidden layer and the regularization parameters between layers are modified between experiments in order to study their influence on the accuracy and performance. The scheme used for notation is defined as follows: $\mathbf{IN} - \mathbf{H_1} - \cdots - \mathbf{H_k} - \mathbf{OUT}$ with $\mathbf{IN}$ the number input nodes for each sample, $\mathbf{OUT}$ the number of output nodes/classes and $\mathbf{H_1} - \cdots - \mathbf{H_k}$ the depths of each hidden layer.

Ten rounds are performed on each topology. Three average parameters are considered to evaluate the performance of each explored ML-ELM. Recognition rate, test time and training time. Their standard deviations are also displayed.

Observing the data displayed in Table I, recognition rates higher than 91.65 % can be achieved. Analyzing the results, it can be stated that increasing the number of hidden neurons in certain layers increases the recognition rate, but for the same topology, the adjustments of the ridge parameters for each layer play an even more important role on increasing the number of hits.

In order to compare the Speed Limit Signs recognition rates with already existing solutions, the GTSRB benchmark results [28] are consulted and shown in Table II.

Analyzing the results of the GTSRB benchmark on Table II and considering that the highest recognition rate achieved in this work is 91.65 %, the proposed Speed Limit Sign Recognition System would surpass the 8th classified of the GTSRB benchmark in performing this task.

As an example of successful recognition under overlighting conditions, the ML-ELM with the topology of the 4th row in Table I is selected. The image in Figure 6 is input to the chosen ML-ELM and the classification outputs are shown in Table III. As it can be observed and remarked in bold characters in Table III, the output with the highest value of 0.98 corresponds with the 70 km/h speed limit sign class. The values that the ML-ELM assigns to the other classes are below -0.91, namely very far from 0.98, which means that not only does the ML-ELM identify the Speed Limit Sign correctly, but also it does it confidently.

Concerning testing times, the more nodes are added to the architecture, the more time it takes to classify the training data. As it can be observed in Table I the biggest network takes almost twice as long as the smallest one since it has

TABLE II

GTSRB SPEED LIMIT SIGNS CLASSIFICATION RATES

Ranking	Solution	Speed Recognition Rate (%)
1	CNN with 3 Spatial Transformers [23]	99.69
...	...	...
7	LDA on HOG 2 [29]	95.37
8	**LDA on HOG 1 [29]**	**91.44**

Fig. 6. A 28×28 B/W bitmap of a 70 km/h Speed Limit Sign suffering from overlighting and bad digit integrity is used as example.

TABLE III
CLASSIFICATION OUTPUTS OF A ML-ELM FOR THE 70 KM/H SIGNAL IN FIGURE 6.

Class:	30	50	60	**70**	80	100	120
Output:	-1.07	-0.91	-0.97	**0.98**	-0.99	-0.97	-1.05

approximately twice as many neurons. Thus, the smallest times happen for the 784-2050-3200-7000-7 topology, no matter the ridge parameters, delaying between 2.59 and 2.55 s to classify all the samples. With these times, and considering that the testing dataset contains 4110 samples, a performance of 0.62 ± 0.17 ms per sample is achieved.

Observing the testing and training times for the biggest network, and taking into account that with properly tuned ridge parameters the recognition rates achieved for smaller networks are pretty similar, it is discarded since its performance does not justify its computational cost.

In addition, comparing Circle Hough Transform extraction times with Classification times, it can be stated that the time it takes to compute this transform (12±0.87 ms per sample) is more than 17 times the computing period of the ML-ELM classifier (0.62 ± 0.17 ms per sample), so the minimum operation time is restricted by the Circle extraction and coincides with it.

V. CONCLUSION AND FUTURE WORKS

In this work, an alternative method to Convolutional Neural Networks for speed limit sign recognition is presented. ML-ELMs present simpler, more understandable topologies than CNNs. The classification process is carried out by a three-hidden-layer ML-ELM. This ML-ELM also performs successfully the extraction of the features since it is a stack of ELM-AEs.

In the line of the results shown in the preceding section, let us note that the recognition rates are achieved by performing no preprocessing apart from a simple contrast normalization with the remarkable fact that the recognized speed limits signs show different lighting conditions, motion blur, digit misplacement, bad integrity and/or partial occlusions.

The results obtained are promising since both Signal Extraction and Feature Extraction and Classification are suitable for Real-Time Processing even using a CPU. Nevertheless, as a future line of work, algorithms, specially Circle Hough Transform might be ported to Graphical Processing Unit (GPU) code in order to take advantage of the parallel, multi-thread processing features of these devices and achieve extremely low working times.

REFERENCES

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, winter 2014.

[2] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool, "Traffic sign recognition x2014; how far are we from the solution?" in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug 2013, pp. 1–8.

[3] M. S. Young and N. A. Stanton, "38 automotive automation: effects, problems and implications for driver mental workload," *Engineering Psychology and Cognitive Ergonomics: Volume 1: Transportation Systems*, p. 347, 2017.

[4] M. Cunningham and M. Regan, "Driver distraction and inattention in the realm of automated driving," *IET Intelligent Transport Systems*, 2017.

[5] L. Kasun, H. Zhou, G.-B. Huang, and C.-M. Vong, "Representational learning with elms for big data," *IEEE Intelligent Systems*, vol. 28, pp. 31–34.

[6] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 920–933, April 2017.

[7] J. Kaur and A. Sharma, "Review paper on edge detection techniques in digital image processing," in *7th International Conference on Innovative Research in Engineering Science and Management (ICIRESM-16) at (IETE), Lodhi Road, New Delhi, India on*, vol. 13, 2016, pp. 168–171.

[8] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, Apr 1988.

[9] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International journal of image processing (IJIP)*, vol. 3, no. 1, pp. 1–11, 2009.

[10] M. Dusio, M. A. Olsena, and C. Busch, "Fingerprint sample quality assessment via ridge line count using laplacian of gaussian edge finding," in *Biometrics and Forensics (IWBF), 2014 International Workshop on*. IEEE, 2014, pp. 1–6.

[11] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[12] 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*, vol. 13, no. 4, pp. 1484–1497, Dec 2012.

[13] M. Á. García-Garrido, M. Á. Sotelo, and E. Martín-Gorostiza, "Fast road sign detection using hough transform for assisted driving of road vehicles," in *Computer Aided Systems Theory – EUROCAST 2005*, R. Moreno Díaz, F. Pichler, and A. Quesada Arencibia, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 543–548.

[14] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, no. EPFL-CONF-227439, 2017.

[15] A. Soetedjo and I. K. Somawirata, "Speed limit traffic sign classification using multiple features matching," in *IT Convergence and Security 2017*, K. J. Kim, H. Kim, and N. Baek, Eds. Singapore: Springer Singapore, 2018, pp. 210–217.

[16] S. K. Berkaya, H. Gunduz, O. Ozsen, C. Akinlar, and S. Gunal, "On circular traffic sign detection and recognition," *Expert Systems with Applications*, vol. 48, pp. 67 – 75, 2016.

[17] B. Mohamed, A. Issam, A. Mohamed, and B. Abdellatif, "Ecg image classification in real time based on the haar-like features and artificial neural networks," *Procedia Computer Science*, vol. 73, pp. 32–39, 2015.

[18] 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 *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug 2013, pp. 1–8.

[19] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Robust neighborhood preserving projection by nuclear/l2,1-norm regularization for image feature extraction," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1607–1622, April 2017.

[20] F. Moutarde, A. Bargeton, A. Herbin, and L. Chanussot, "Robust on-vehicle real-time visual detection of american and european speed limit signs, with a modular traffic signs recognition system," in *2007 IEEE Intelligent Vehicles Symposium*, June 2007, pp. 1122–1126.

[21] S. L. Gomes, E. d. S. Rebouças, E. C. Neto, J. P. Papa, V. H. C. d. Albuquerque, P. P. Rebouças Filho, and J. M. R. S. Tavares, "Embedded real-time speed limit sign recognition using image processing and machine learning techniques," *Neural Computing and Applications*, vol. 28, no. 1, pp. 573–584, Dec 2017.

[22] Y. Liu, D. Yang, and C. Zhang, "Relaxed conditions for convergence analysis of online back-propagation algorithm with l2 regularizer for sigma-pi-sigma neural network," *Neurocomputing*, vol. 272, pp. 163 – 169, 2018.

[23] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Networks*, vol. 99, pp. 158 – 165, 2018.

[24] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489 – 501, 2006.

[25] 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*, July 2011, pp. 1453–1460.

[26] S. J. K. Pedersen, "Circular hough transform," *Aalborg University, Vision, Graphics, and Interactive Systems*, vol. 123, p. 123, 2007.

[27] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," *International journal of computer science and management Research*, vol. 2, no. 2, pp. 1578–1583, 2013.

[28] "Gtsrb benchmark results." [Online]. Available: http://benchmark.ini.rub.de/index.php?section=gtsrb&subsection=results&subsubsection=official#

[29] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323 – 332, 2012, selected Papers from IJCNN 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012000457