EFFICIENT LANE DETECTION BASED ON ARTIFICIAL NEURAL NETWORKS

Fernando Arce [a], Erik Zamora [b], Gerardo Hernández [a], Humberto Sossa [a,*]

[a] Instituto Politécnico Nacional, CIC, Av. Juan de Dios Bátiz S/N, Col. Nueva Industrial Vallejo, Gustavo A. Madero, 07738, Ciudad de México, México
[b] Instituto Politécnico Nacional, UPIITA, Av. Instituto Politécnico Nacional 2580, Col. Barrio la Laguna Ticomán, 07340, Ciudad de México, México

KEY WORDS: Artificial Neural Networks, Ellipsoidal Neuron, Dendritic Processing, Lane Detection.

ABSTRACT:

Lane detection is a problem that has attracted in the last years the attention of the computer vision community. Most of approaches used until now to face this problem combine conventional image processing, image analysis and pattern classification techniques. In this paper, we propose a methodology based on so-called Ellipsoidal Neural Networks with Dendritic Processing (ENNDPs) as a new approach to provide a solution to this important problem. The functioning and performance of the proposed methodology is validated with a real video taken by a camera mounted on a car circulating on urban highway of Mexico City.

1. INTRODUCTION

Autonomous Car Technology (ACT) is recognized nowadays as one of the most important advances in urban mobilization. It is under development since 15 years. Between 2004 and 2005 several vehicles competed to traverse the Mojave Desert in California covering more than 200 km in an autonomous way, following a map of GPS coordinates. The vehicles were built by universities such as Stanford and Carnegie Mellon, as well as by industries like Ensco and Velodyne Acoustics. The event was organized by DARPA-USA. In 2007, another competition was organized. It represented a major challenge: Vehicles were asked to avoid hitting other vehicles, cycles, pedestrian or traffic lights. They should also have the capacity to make "U" turns, when needed, and to transit under different weather conditions. In 2005 and 2007 the two winners were Stanley (Thrun, 2006) and Boss (Urmson, 2008), respectively.

Many approaches for automatic lane detection have been reported in the literature: (Assidiq 2008), (Fernando 2014), (Saha 2012), (Van Winden 2016) and (Wang 2006), to mention a few.

In (Assidiq 2008), for example, the authors solve lane detection by first acquiring a front view by a camera mounted on the vehicle. They then detect lanes by using a pair of hyperbolas which are fitting to the edges of the lane by means of so-called Hough transform. They show that their proposal is robust to light changes and shadows by testing it with both painted and unpainted road as well as curved and straight road in different weather conditions.

In (Fernando 2014), the authors face lane detection by integrating two visual cues: 1) stripe-like features found on lane lines extracted using a two-dimensional symmetric Gabor filter, and 2) texture characteristic determined using the entropy measure of a predefined neighbourhood around a lane boundary line. They integrate both visual cues using a rule-based classifier which incorporates a modified sequential covering algorithm to improve robustness. They separate lane boundary lines from other similar features by using a road mask that uses road

chromaticity values estimated from CIE L*a*b* colour transformation. They validate their algorithm on a set of images of suburban streets.

In (Saha 2012), the authors face the lane detection problem by first converting a RGB road scene image into a grey image. They then employ a flood-fill algorithm to label the connected components on the grey image. After that, they take the largest connected component as the road region. They finally subtract the marks or road lane and road boundary from the connected components. Thy show the performance of their algorithm on both straight and slightly curved road scene images under different day light conditions and the presence of shadows on the roads.

In (Van Winden 2016), the authors describe a method for automatically deriving road attributes by analysing and mining movement trajectories (e.g. GPS tracks). They investigate how to automatically extract eight road attributes: directionality, speed limit, number of lanes, access, average speed, congestion, importance, and geometric offset. For this, they propose using a decision tree. They use their proposal to automatically update the OpenStreetMap (OSM) dataset of the Netherlands, increasing its level of completeness.

In short, in (Wang 2006), the authors face real-time lane detection by combining a hyperbola-pair lane boundary model and an improved RANSAC paradigm. They introduce a fuzzy measurement into the RANSAC paradigm to improve accuracy and robustness when fitting boundaries points into the model. They test their proposal in many different conditions, including various conditions of illumination, weather and road.

It is worth mentioning that most lane detection methods are based on the Hough transform (Leng, 2010), (Li, 2014), (Tan, 2014), (Gaikwad, 2015), a well-known technique used to detect lines and curves (Duda, 1972). This kind of line detectors strongly depend on the pre-processing stage; they are typically combined with a Canny or Sobel edge detector or with an adaptive threshold. The main drawback is in the complexity of the

---

*   Corresponding Author. Tel.: +52 55 5930 6193.
    E-mail: humbertosossa@gmail.com (Humberto Sossa).

methodology and the required time. Our proposed does not use these filters and is so easy to implement.

Most recent approaches make use of so-called deep networks (Kim, 2014), (Huval, 2015), (Gurghian, 2016). These techniques are so accurate and represent the state-of-the-art in image classification. They have three main disadvantages: need 1) thousands of examples for training, 2) specialized hardware to be implemented and 3) are time consuming. The besought method does not require a special hardware neither is time consuming.

To end up with this discussion, in (Kaur 2015), the authors provide a comprehensive review of the literature in lane detection techniques with the aim to present to the reader the limitations of the existing lane detection methods.

Also it is expected that in 2035, the more developed societies will adopt ACT as a normal transportation way. While that moment arrives, several intelligent vehicular assistants are under development. One of these systems concerns automatic lane detection. Systems like these alert the driver when his car starts to move out of the lane; either because the driver is distracted or tired. A system like this is particularly useful in highways where monotony of conduction can augment distraction.

Artificial neural networks, on the other hand, have been successfully used in many tasks including signal analysis, noise cancellation, model identification, process control, object detection, and pattern recognition, in general.

Many artificial neural network models have been reported in literature, since the very simple Threshold Logic Unit (TLU), introduced by McCulloch-Pitts (MuCulloch 1943) at the beginning of the 40´s, passing by the well-known Perceptron, presented to the world by Rosenblatt in the 50´s (Rosenblatt 1958) and (Rosenblatt 1962) until the so called Morphological Neural Models with and without Dendritic Processing introduced by Ritter et al. in (Ritter 1996), (Sussner 1998), (Ritter 1999), (Ritter 2003), to mention a few.

Interest in Morphological Neural Networks (MNNs) has increased in the last five years (Sossa, 2013), (Sossa, 2014), (Arce, 2016) and (Zamora, 2017). This kind of ANNs make use of **min** and **max** operations to segment the input space into hyper-boxes. In particular, MMNs with dendritic processing have shown the capacity to efficiently solve low-dimensional classification problems where other reported methods fail (Zamora, 2016), (Hernandez, 2017).

In this paper, we propose a new method for lane detection based on the functioning of so-called dendritic ellipsoidal neurons (DEN) (Arce, 2017). Instead of hyper-boxes, these neural networks use hyper-ellipsoids.

The rest of the paper is organized as follows. In Section 2 we briefly present the architecture of an ellipsoidal neuron, while in Section 3 we describe, step by step, how this kind of neuron can be trained. Section 4, in the other hand, is focused to introduce the method for detecting lanes in an image. In Section 5 we present some results, while in Section 6 we summarize the conclusions and directions for further research.

2. ARQUITECTURE OF A DEN

Before presenting the DEN, we describe the conventional mathematical structure for a Dendrite Morphological Neuron (DMN). The dendrite output for a DMN is given by:

$$\tau_k^j = \bigwedge_{i=1}^{n} \left(x_i + w_{ik}^1 \right) \wedge - \left(x_i + w_{ik}^0 \right) \qquad (1)$$

where x_i is one of the n components of an input vector of $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. w_{ik}^0 and w_{ik}^1 are the synaptic weights (w_{ik}^0 inhibition, w_{ik}^1 activation) of terminal fibres of the ith neuron synapsing the kth dendrite for the jth class. More details can be found in (Ritter 2003) and (Ritter 2006).

The neural network model used in this paper also uses dendritic processing (Arce 2017), the authors call it as Dendrite Ellipsoidal Neuron (DEN). Instead of using hyper-boxes to cluster data, as normally used by standard DMNs, DENs use hyper-ellipsoids. Next we describe each of the parts of a DEN. This change is key for our lane detector because the hyper-boxes cannot be rotated. The hyper-boxes in the DMNs are always align to the coordinate axes which is a great limitation in the moment to segment the lane. In contrast, the hyper-ellipsoids have the flexibility to be rotated. This allows to need less number of dendrites to segment the same number of pixels for the lanes. Of course, everything has a cost. The DEN requires $n(n + 1)$ number of learning parameters per dendrite, while the DMNs requires $2n$ parameters per dendrite, where n is the dimension of patterns. Since the lane detection is a classification problem with 2 dimensions (for black and white images), this is not important for the lane detector.

The architecture of a DEN is illustrated in Figure 1. As can be seen, it is composed of three layers as follows:

2.1 Input Layer

This layer receives at its input n-dimensional vectors of the form $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. This layer does not perform any processing, it only sends the information provided by the input patterns to the next layer, the so-called dendritic layer.

2.2 Dendritic Layer

This layer performs so-called dendritic computation over input patterns $\mathbf{x}_i, i = 1, \ldots, m$, by applying the following equation:

$$\tau_k^j = (\mathbf{x}_i + \mathbf{\mu}_k)^T \sum_k^{-1} (\mathbf{x}_i + \mathbf{\mu}_k) \qquad (2)$$

In this case, each variable is defined as follows:

τ_k^j = Dendritic computation	T = Transpose
$\mathbf{x}_i$ = Input pattern	j = Index of class C^j
$\mathbf{\mu}_k$ = Centroid of ellipse k	k = Dendrite
$\sum_k^{-1}$ = Covariance matrix	m = Number of training patterns

τ_k^j is the vector produced by all the dendrites; it represents the Mahalanobis distance between input pattern $\mathbf{x}_i$ and all the generated ellipsoids. Each dendrite performs the operation given by Equation (2) on each input pattern. The number of dendrites as well as the number of centroids and covariance matrices are obtained during the training step.

2.3 Output Layer

This layer is composed of only one neuron. Once the dendritic step is performed, the output neuron choses the dendrite that has

generated the smallest distance between the input pattern and all generated ellipses; it then assigns the input pattern to the corresponding class. The following equation shows how this calculation is made:

$$\tau_j = argmin_k(\tau_k^j) \qquad (3)$$

τ_j is a scalar value that indicates the nearest distance between the input pattern and all the placed ellipsoids.

Given an input pattern: $\mathbf{x}_i$, if $\tau_j = 0$, means that $\mathbf{x}_i$ is exactly at the centroid of the hyper-ellipse. If $\tau_j < 1$, $\mathbf{x}_i$ is inside the hyper-ellipse, and if $\tau_j > 1$, $\mathbf{x}_i$ is out of the hyper-ellipse.

Figure 1 shows the architecture of an ellipsoidal neuron with dendritic processing. Figure 2 shows an example (in 2D) of a hyper-ellipse generated by its two dendritic parameters ($\boldsymbol{\mu}_k$ and $\sum_k^{-1}$).
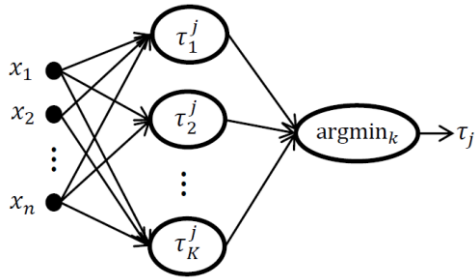

Figure 1. Architecture of an ellipsoidal neuron with dendritic processing, composed of three layers. Taken from (Arce 2017).
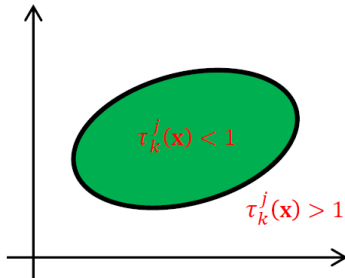

Figure 2. Example of a hyper-ellipsoid generated by the two parameters ($\boldsymbol{\mu}_k$ and $\sum_k^{-1}$) of the dendrite. Taken from (Arce 2017).

3. TRAINING OF A DEN

One of the most important problems in training a neural network with dendritic processing is the selection of the number of dendrites. The method described in this note automatically determines the number of necessary dendrites to solve a given problem. In this section we explain the operation of the proposed algorithm. Before beginning, let us have the following definitions:

Definition 1. For $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m, i = 1,2, ..., m$ a finite set of patterns; $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{in})^T \in \mathbb{R}^n$ being one of these patterns. Each pattern $\mathbf{x}_i$ belongs to only one of p classes: $C^j, j = 1,2, ..., p$, with $p > 1$ a finite number.

Definition 2. A hyper-ellipsoid HE^n is an n-dimensional ellipsoid that contains a finite set of patterns: $\mathbf{x} \in \mathbb{R}^n$. This HE^n is defined by its dendritic parameters: ($\boldsymbol{\mu}_k$ and $\sum_k^{-1}$).

Equation 2 defines the Mahalanobis distance between pattern $\mathbf{x}_i$ and hyper-ellipsoid HE^n. The goal of the proposed training algorithm is to create and position a set of hyper-ellipsoids $HE_K^N \in \mathbb{R}^{K \times n}$, with $K \in \{1,2, ..., k\}$ that allow stablishing a decision surface among classes with the smallest number of dendrites.

3.1 Steps of the Training Algorithm

1) Define a global error ($error_{global}$) and local error ($error_{local}$). Both errors are referred to the allowed number of wrong classified patterns. $error_{global}$ is the total allowed error while $error_{local}$ makes reference to the allowed errors (positive failures and negative failures) for each class.

2) For each class C^j, generate k clusters using so-called k-means++ algorithm (Arthur, 2007), $k \in \mathbb{Z}^+$.

    a. Obtain the dendritic parameters ($\boldsymbol{\mu}_k$ and $\sum_k^{-1}$) of each cluster k. Inverse of the covariance matrix can be obtained by applying the pseudo-inverse Moore-Penrose method.

    b. Compute $error_{local}$ counting the number of wrong classified patterns.

    c. If $error_{local} < error_{global}$ continue with the following class and go back to step 2. Else, $k = k + 1$.

Algorithm 1 depicts the training for an ellipsoidal neuron. Note that it uses counter c to avoid falling in an infinite cycle when no better classification improvement is obtained.

```
BEGIN
Define error_global y error_local
FOR each class C^j, k = 1:
    WHILE error_local > error_global:
        Transform the k clusters into hyper-ellipsoids
        Compute error_local
        IF error_local == 0:
            STOP
        IF error_global ≥ |error_local[k − 1] − error_local[k]|:
            c = c + 1
            IF c == 3
                STOP
        k = k + 1
END
```

Algorithm 1. Training algorithm for a DEN.

To better understand the functioning of a DEN, the following straightforward example is provided. As can be seen from Figure 3, it is two class problem: The blue dots belonging to the first class C^1 and the green stars to the second class C^2.

We would like to classify pattern $\tilde{\mathbf{x}}_1 = [-1.0000 \; -0.5000]$ which is supposed to belong to class C^1 and pattern $\tilde{\mathbf{x}}_2 = [0.0000 \; 0.0000]$ from class C^2. Table 1 presents the hyper-parameters of the two generated ellipsoids by Algorithm 1.
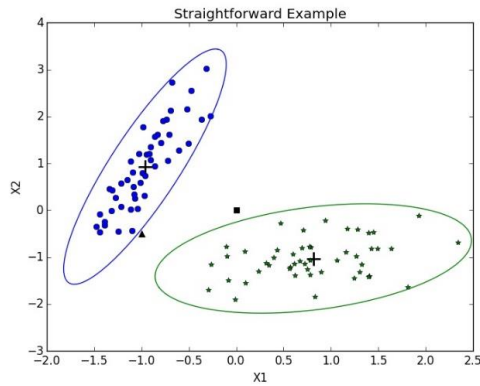
Figure 3. DEN trained to solve a straightforward example. Blue ellipsoid encloses patterns belonging to first class; green ellipsoid does the same but patterns from second class.

Table 1. Hyper-parameters of the trained DEN to solve the provided straightforward example.

Hyper-parameters:	
$\boldsymbol{\mu}_1 =$	$[-0.974 \quad 0.9325]$
$\sum_k^{-1} =$	$\begin{bmatrix} 0.0985 & 0.2328 \\ 0.2328 & 0.7862 \end{bmatrix}$
$\boldsymbol{\mu}_2 =$	$[0.8518 \ -1.0341]$
$\sum_2^{-1} =$	$\begin{bmatrix} 0.42111 & 0.0672 \\ 0.0672 & 0.1722 \end{bmatrix}$

When Equation (2) is applied to both input patterns we obtained, we can show that:

$$\boldsymbol{\tau}_1^j = [3.2309 \ 20.1368]$$

and

$$\boldsymbol{\tau}_2^j = [19.0885 \ 14.0108]$$

respectively.

Finally, by applying Equation (3) to these dendrite computations $\boldsymbol{\tau}_k^j$ and we obtain:

$$\tau^1 = argmin([3.2309 \ 20.1368]) = 1$$

And

$$\tau^2 = argmin([19.0885 \ 14.0108]) = 2.$$

Therefore pattern $\tilde{\mathbf{x}}_1 = [-1.0000 \ -0.5000]$ is classified into class C^1, while pattern $\tilde{\mathbf{x}}_2 = [0.0000 \ 0.0000]$ is put into class C^2. Both noisy patterns are correctly classified as expected.

4. LANE DETECTION USING A DEN

In this section we describe the methodology for lane detection based on ellipsoidal neurons with dendritic processing. The experiment was performed by using a video taken at the "Viaducto Bicentenario" of the State of Mexico. The video consists in 3237 frames with a resolution of 580×620 pixels per frame. We used low resolution images due to we are focused to implement our proposal in a near future in an embedded device. The methodology was applied to each frame and uses the following four steps:

4.1 Change of Colour Space

The first step consists in changing the colour space from RGB to HLS. The idea is to work in the S cannel (Saturation) that provides the degree at which a primary colour is dissolved (yellow, orange, or pure red) pure with light (Gonzalez, 1987). Figure 4 shows an original image while Figure 5 depicts the S cannel of the HLS colour space. In Figure 5 we can observe the slight improvement in colour on the image lanes.

Figure 4. Original imagen taken at the "Viaducto Bicentenario" of the State of Mexico.

Figure 5. Saturation channel S of colour space HLS of the original image.

4.2 Selection of the Interest Area

The following step consists in selecting an interest area in the image. The idea is to isolate from the S-image an area containing, with a high probability, the lanes we would like to further detect. Figure 6 depicts an example of interest area.

Figure 6. Selection of the area of interest.

The interest area is defined to have a trapezoidal form, taking into account the knowledge that at the distance lanes tend to join. The lower corners of the trapezoid are chosen as the image corners to the left and to the right of the image. The upper corners of the trapezoid are marked by using the perspective information of the camera.
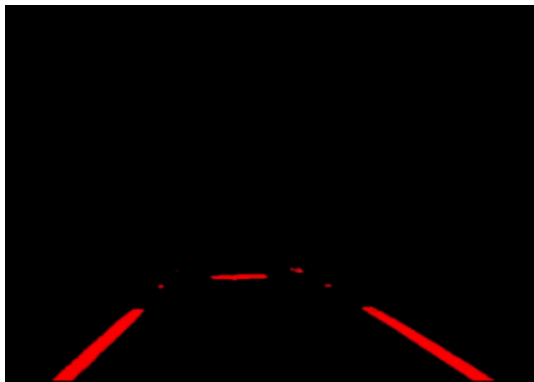

Figure 7. Thresholding of the interest area.

4.3 Interest Area Thresholding

Once the interest area has been selected, the proposed methodology proceeds to threshold it. Two thresholds were used: 50 and 120, respectively. Figure 7 presents the thresholded interest area. As can be appreciated this area contains interest pixels (in red) and non-interesting pixels (in black).
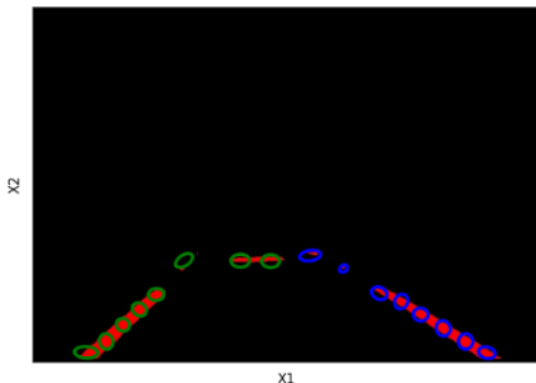

Figure 8. Establishment of the ellipses on the binary image by means of an ENNDP.

4.4 Application of the ENNDPs to The Binary image

The following step consists in stablishing ellipses over the thresholded image by means of the trained ellipsoidal neuron. To carry out this, during training the red pixels were assigned to a first class (class "0") and the black pixels to the second class (class "1"). Instead of assigning an optimal number of ellipses as it will done when applying Algorithm 1, a constant number of 16 ellipses per frame were marked. This allows reducing computation times. Figure 8 shows the ellipses generated by the proposal using the ellipsoidal neuron.

4.5 Ellipse Filtering

The last step of the methodology for automatic lane detection consisted in filtering ellipses by their centroids, created in advance by means of RANSAC method. This random algorithm is normally used, as it is known, to eliminate atypical values not

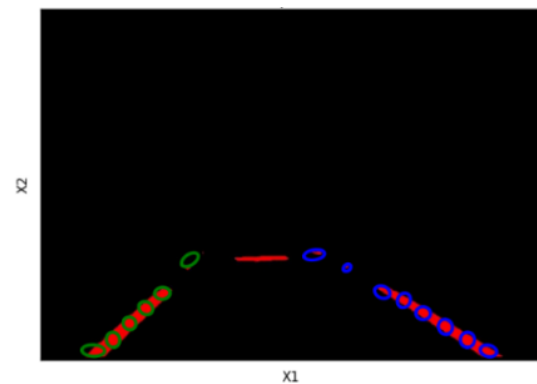coinciding with the model (Fischler, 1981). Figure 9 shows the filtered ellipses using RANSAC.

Figure 9. Ellipse filtering using RANSAC.

5. EXPERIMENTAL RESULTS

As already mentioned in Section 4, the described methodology was applied on a video composed of 3237 frames with a resolution of 580×620 pixels per frame, taken with camera mounted on a car circulating on "Viaducto Bicentenario", State of Mexico. Figure 9 shows nine of the results obtained by the application of the described methodology.
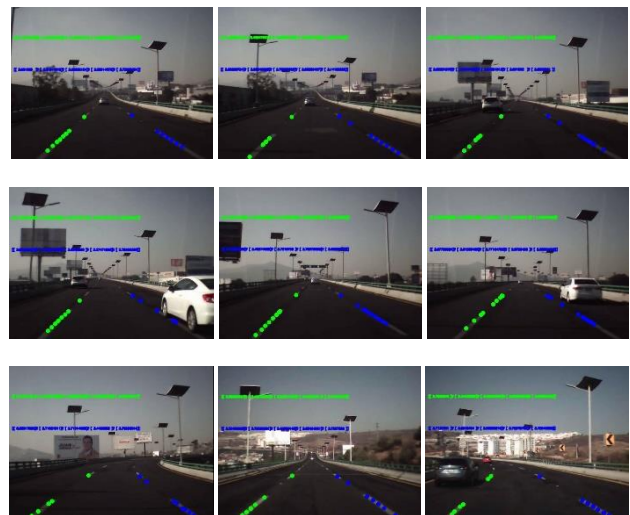

Figure 9. Nine of the images with the ellipses superimposed by the algorithm over the lanes.

It is worth mentioning to say that in 74 frames the algorithm fails to correctly superimpose the ellipses. The calculated accuracy of the proposed methodology over this set of images was thus of 0.977.

The main drawback of the proposed algorithm is the required time by the filtering stage; it consumes 40% of the total processing time. A possible solution could be using other filtering techniques and an implementation of the algorithm in a Graphic Processing Unit (GPU).

6. CONCLUSIONS AND FUTURE WORK

We have presented a methodology based on the pattern recognition capacities of the so-called dendrite ellipsoidal

neuron. We have shown how the proposed methodology can be successfully applied to automatically detect lanes in urban highways.

Future work includes the implementation of the besought algorithm on a portable Graphic Processing Unit (GPU) and the execution of this on a scaled exploration car. Another future work will be the use of other filtering techniques focused in the time reduction in this stage.

Finally, a formal evaluation and a deep analysis of the proposed method are necessary to validate its whole capacities compared to other methods and algorithms reported in literature.

ACKNOWLEGMENTS

REFERENCES

Arce, F. Zamora, E., Sossa, H. and Barrón, R., 2016. "Dendrite Morphological Neural Networks trained by Differential Evolution," 2016 IEEE Symposium Series on Computational Intelligence (SSCI 2016), Athens. pp. 1-8.

Arce, F., Zamora, E. and Sossa, H., 2017. "Dendrite Ellipsoidal Neuron", International Joint Conference on Neural Networks (IJCNN 2017), USA, May 14-19th.

Arthur, D. and Vassilvitskii, S., 2007. K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 1027–1035.

Assidiq, A., Khalifa, O. O., Islam, Md. R. and Khan, S. 2008. "Real Time Lane Detection for Autonomous Vehicles". International Conference on Computer and Communication Engineering (ICCCE 2008). Kuala Lumpur, Malaysia. Pp. 82-88.

Duda, R. O. and P. E. Hart, 1972. "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15.

Fernando, S., Udawatta, L., Horan, B. and Pathirana, P., 2014. "Real-time Lane Detection on Suburban Streets using Visual Cue Integration". International Journal of Advanced Robotic Systems 11:1-20.

Fischler, M. A. and Bolles, R. C., 1981. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography". Communications of the ACM 24(6), pp. 381–395.

Gaikwad, V. Lokhande, S., 2015). "Lane departure identification for advance driver assistance", IEEE transaction on intelligent transportation system, vol.16, no. 2. pp. 1-9.

Gonzalez, R. C. and Wintz, P., 1987. Digital Image Processing (2Nd Ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Gurghian, A., et al, 2016. "DeepLanes: End-To-End Lane Position Estimation Using Deep Neural Networks". CVPR Workshops 2016: pp.38-45.

Hernandez, G. Zamora, E, and Sossa, H., 2017. "Comparing Deep and Dendrite Neural Networks: A case Study", Mexican Conference on Pattern Recognition (MCPR 2017), México, June 21-24th.

Huval, B. Wang T., Tandon ,S. et al., 2015. "An empirical evaluation of deep learning on highway driving," https://arxiv.org/abs/1504.01716.

Kaur, G. and Kumar, D., 2015. "Lane Detection Techniques: A Review". International Journal of Computer Applications 112(19):4-8.

Kim, J. Lee, M., 2014. "Robust Lane Detection Based On Convolutional Neural Network and Random Sample Consensus". C.K. Loo et al. (Eds.): ICONIP 2014, Part I, LNCS 8834, pp. 454–461.

Leng, Y. C. Chen, C. L., 2010. "Vision-based lane departure detection system in urban traffic scenes," in Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on, pp. 1875–1880.

Li, Y., Iqbal, A., & Gans, N. R., 2014. "Multiple lane boundary detection using a combination of low-level image features". IEEE 17th International Conference on Intelligent Transportation Systems (ITSC2014), pp. 1682-1687.

MuCulloch, W. S. and Pitts, W. H., 1943. "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133.

Ritter, G. R. and Sussner, P., 1996. "An introduction to morphological neural networks," Proceedings of the 13th International Conference on Pattern Recognition, 1996, vol. 4, pp. 709–717.

Ritter, G. R. and Beaver, T. W., 1999. "Morphological perceptrons" International Joint Conference on Neural Networks, vol. 1, pp. 605–610.

Ritter, G. R., Iancu, L. and Urcid, G., 2003. "Morphological perceptrons with dendritic structure," 12th IEEE International Conference in Fuzzy Systems (FUZZ 2003), vol. 2, pp. 1296–1301.

Ritter, G. X. and Schmalz, M. S., 2006. "Learning in lattice neural networks that employ dendritic computing," in Fuzzy Systems, 2006 IEEE International Conference on, pp. 7–13.

Rosenblatt, F., 1958. "The perceptron: A probabilistic model for information storage and organization in the brain," Psychological Review, vol. 65, pp. 386–408.

Rosenblatt, F., 1962. "Principles of neurodynamics: Perceptron and theory of brain me- chanisms," Spartan, Washington DC.

Saha, A. Roy, D. D., Alam, T. and Deb, K., 2012. "Automated Road Lane Detection for Intelligent Vehicles". Global Journal of Computer Science and Technology 12(6):1-4.

Sossa, H., Guevara, E., 2013. "Modified Dendrite Morphological Neural Network Applied to 3D Object Recognition". LNCS 7914, Springer-Verlag, pp. 314-324.

Sossa, H. and Guevara, E., 2014. "Efficient training for dendrite morphological neural networks". Neurocomputing 131:132-142.

Sussner, P., 1998. "Morphological perceptron learning," IEEE ISIC/CIRA/ISAS Joint Conference, pp. 477–482.

Tan, H. Zhou, Y. Zhu, Y. Yao, D. Li, K., 2014. "A novel curve lane detection based on improved river flow and RANSA". In Proceedings of the IEEE Conference on Intelligent Transportation Systems, pp. 133–138.

Thrun S., et al., 2006. "Stanley: The robot that won the DARPA Grand Challenge". Journal of Robotic Systems – Special Issue on the DARPA Grand Challenge, Vol. 23, No. 9, pp. 661-692.

Urmson C., et al., 2008. "Autonomous driving in urban environments: Boss and the Urban Challenge". Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I. Vol. 25, No.8, pp. 425-466.

Van Winden et al. (2016). "Automatic Update of Road Attributes by Mining GPS Tracks". Transactions in GIS, 20(5), 664–683.

Wang, H. and Chen, Q., 2006. "Real-time Lane Detection in Various Conditions and Night Cases". IEEE Intelligent Transportation Systems Conference (ITSC 2006). Toronto, Canada, September 17-20, Pp. 1226-1231.

Zamora, E. and Sossa, H., 2016. "Regularized divide and conquer training for dendrite morphological neurons". Mechatronics and Robotics Service: Theory and Applications, Mexican Mechatronics Association, November 2016.

Zamora, E. and Sossa, H., 2017. "Dendrite morphological neurons trained by stochastic gradient descent". Neurocomputing doi.org/10.1016/j.neucom.2017.04.044.