Actively Semi-Supervised Deep Rule-based Classifier Applied to Adverse Driving Scenarios

Eduardo Soares, Plamen Angelov, Bruno Costa, Marcos Castro

Abstract—This paper presents an actively semi-supervised multi-layer neuro-fuzzy modeling method, ASSDRB, to classify different lighting conditions for driving scenes. ASSDRB is composed of a massively parallel ensemble of AnYa type 0-order fuzzy rules. It uses a recursive learning algorithm to update its structure when new data items are provided and, therefore, is able to cope with nonstationarities. Different lighting conditions for driving situations are considered in the analysis, which is used by self-driving cars as a safety mechanism. Differently from mainstream Deep Neural Networks approaches, the ASSDRB is able to learn from unseen data. Experiments on different lighting conditions for driving scenes, demonstrated that the deep neuro-fuzzy modeling is an efficient framework for these challenging classification tasks. Classification accuracy is higher than those produced by alternative machine learning methods. The number of algebraic calculations for the present method are significantly smaller and, therefore, the method is significantly faster than common Deep Neural Networks approaches. Moreover, DRB produced transparent AnYa fuzzy rules, which are human interpretable.

I. INTRODUCTION

In recent years, driven by the vast and exponentially increasing amounts of data, there has been a significant advancement of artificial intelligence (AI) [1], [2]. Improvements in both algorithms and powerful computer hardware allowed AI to enter into a new developmental stage [3]. Deep Learning neural networks (DLNNs), have emerged from these advancements and have become the state-of-the-art in machine learning field [4]. As demonstrated in [5], DLNNs can produce highly accurate results in problems from various knowledge domains, as they are able to map intricate structures in high-dimensional data into learning weights. DLNNs are essential for solving problems that have been confronting the artificial intelligence community for many years, specially in research involving self-driving cars. This area directly benefited from these advancements, as it depends on highly accurate systems to operate safely [6].

However, DLNNs suffer from a number of shortcomings, as they depend on a huge number of labeled training data, learn in offline mode and require many decisions to be made by the user, such as threshold selection, algorithmic parameters, number of layers, and other *ad hoc* choices [7]. It is also well-known that DLNN-based approaches are not able to cope with uncertainty and are usually not equipped with incremental online algorithms capable of adapting their

Eduardo Soares and Plamen Angelov are with the School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK. Bruno Costa and Marcos Castro are with the Ford Research and Innovation Center, Ford Motor Company, Palo Alto, CA 94304, USA. E-mails:e.almeidasoares@lancaster.ac.uk; p.angelov@lancaster.ac.uk; bcosta17@ford.com; mgerard8@ford.com.

parameters and structure. DLNN-based approaches are able to perform classification tasks with high accuracy rate when the validation images share similar features properties with the training images. However, substantial amounts of time, computer memory and historical well-distributed-over-the-problem-domain data are required for parameter adjustment. Similarly to typical statistical models [8], offline designed DLNNs do not cope with concept changes and novelties [9].

A promising approach to deal with large amounts of data is through the use of a class of methods known as evolving intelligent systems [10-14]. Evolving intelligent systems introduced in [15], offer an effective way of handling nonstationary data streams since models become able to adapt their parameters and structure to different situations and track environmental changes, i.e., concept drifts and shifts [16].

A challenging topic in the field of self-driving cars concerns classification of unseen scenes. Difficulties arise mainly because self-driving cars requires an extraordinarily high degree of accuracy and reliability considering environment perception [17]. Therefore, mechanisms that can provide a guarantee of safety is of supreme importance to public acceptance of autonomous vehicles [18]. Moreover, if autonomous vehicles are involved in any accident it is of paramount importance to know the exact reasons. Consequently, systems need to provide transparency, interpretability, and explainability in order to support specialists and be acceptable by the public [19].

This paper presents a multi-layer neuro-fuzzy model called Deep Rule-Based (DRB) system [9], [20]. The DRB approach has as a learning engine composed of a massively parallel set of 0-order fuzzy rules, which are able to self-adapt and provide transparent and human understandable *IF ... THEN* representation [21]. It is also able to self-evolve its structure and self-update its meta-parameters as newly observed training images arrive from the data stream, which makes the classifier applicable for real-time applications [9], [20].

In this paper, the DRB classifier is automatically designed and is applied to unseen scenes from adverse driving conditions. The hypothesis is that the DRB model can work as a safety mechanism, in order to support DLNN in unexpected situations, as a scene never seen can cause problems if the DLNN was not trained to recognize it. Results are compared in the sense of accuracy, model compactness and processing time. The main contributions of this paper are:

- It offers a new method to automatically and reliably detect unseen scenes (low confidence in any of the known classes).

- A mechanism to automatically learn new unseen scenes and in a form of active learning label them (optionally) to provide human-understandable meaning.
- A human-interpretable, computationally efficient classifier outperforming the competitors.

The remainder of this paper is structured as follows. Section II presents the Deep Rule-Based evolving classifier approach. Section III describes the methodology employed in the analysis. Results and discussions are shown in Section IV. Conclusion and future research directions are given in Section V.

II. Deep Rule-Based Classifier

DRB is a semi-supervised multi-layer neuro-fuzzy modeling method that produces transparent and human understandable *IF ... THEN* fuzzy rules based on image prototypes. An incremental learning algorithm autonomously self-develops its rule structure to track new concepts, cope with uncertainty, self-evolve and update its structure and meta-parameters if and when unseen classes are presented to the system, which makes it very suitable for real-time and autonomous applications. As shown in Figure 1, DRB is composed of the following layers: 1) Scaling Layer; 2) Normalization Layer; 3) Feature Descriptor; 4) Fuzzy-Rule base Layer; and 5) Decision Making Layer.
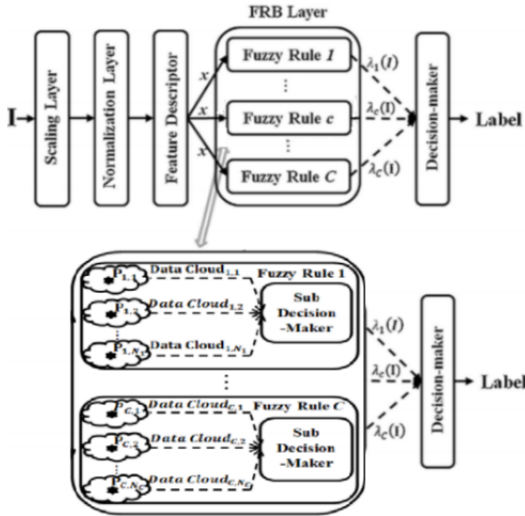

Fig. 1. Deep Rule-Based architecture [7]

1) **Scaling layer**: (Optional)
   The Scaling Layer is responsible to resize original images in order to reduce computational complexity, increase generalization, etc. Specifically, in this paper all images were resized to 227×227 as required by the feature descriptor.
2) **Normalization Layer:** (A standard pre-processing step)
   The Normalization Layer is responsible to normalize the pixel values of the images into the range required by the feature descriptor. In this paper, pixels are normalized between [0,1], which is the range required by the feature descriptor.

3) **Feature Descriptor Layer**: (Defines the date space)
   The Feature Descriptor Layer is in charge of extracting global feature vectors from the images. Features that are extracted are used for training and validation of the classification model. A widely used technique exploited to extract global features from images is to use pre-trained Deep Convolutional Neural Networks (DCNN), as they are the state-of-art approach in image recognition. Features extracted from the pre-trained DCNN may be used to form the feature/data space in which is build the actual classifier. The use of pre-trained DCNN in this way allows the classification model to consider more abstract and discriminative high-level features; consequently, achieving greater performance.
   In this paper, pre-trained VGG–VD–16 DCNN is employed as a feature extraction. According to [22], VGG–VD–16 has a simple structure and can achieve a better performance in comparison with other pre-trained DC-NNs. The first fully connected layer from VGG–VD–16 provides a 1×4096 dimensional activations vector.
4) **Fuzzy-Rule base layer**: (Model engine)
   The Fuzzy-Rule base (FRB) layer is the main engine of the DRB classifier and is constituted of a massively parallel ensemble of AnYa type 0-order fuzzy rules [23]. This layer is responsible to provide the highly accurate classifier which also offers interpretability and transparent models for human understanding, unlike the mainstream Deep Learning approaches [4], [6], which are 'black box' as they do not provide insights about the structure of the network. As it can be seen in Fig 1, the FRB layer consists of 0-order fuzzy rule-based subsystems of AnYa type [23]. Due to its prototype-based nature, the DRB classifier is free from *prior* assumptions about the data distribution type, as well as the random or deterministic nature of the data.
   The FRB layer is composed of subsystems which are entirely independent from each other. Therefore, it can be changed without influencing other subsystems. Each FRB subsystem contains a set of massively parallel AnYa type 0-order fuzzy rules, trained in parallel one per class corresponding to the classes from the image set. Rules are formulated around the prototypes (which are the local peaks of the data density (and typicality) identified during the training stage from the images of the corresponding class based on their feature vectors. These 0-order AnYa type fuzzy rules have the following form:

$$R_i^c: \text{IF } (I \sim P_i^c) \text{ THEN } (class\ c)$$

where $\sim$ stands for similarity, it also can be seen as a fuzzy degree of membership; $c = 1, 2, ..., C$; N^c denotes the number of prototypes of the cth class; P_i^c is the identified prototypes, $i = 1, 2, ..., N^c$.
Rules can be combined through the maximum aggregation operator S-norm (logical OR, also known as disjunction). Combined fuzzy rule has the following

form per class:

R_i^c: IF $(I \sim P_1^c)$ OR $(I \sim P_2^c)$ OR ... OR $(I \sim P_{N^c}^c)$ THEN $(class\ c)$

In order to determine the overall degree of satisfaction, the local, per class 'winner-takes-all' decision-making principle is applied.

5) **Decision-maker**:

The global decision-maker layer is in charge of forming the overall decision by assigning labels to the validation images based on the degree of similarity of the prototypes obtained by the FRB layer.

A. Training Process

In this subsection the DRB training process is summarized. Identified prototypes in the FRB layer are the peaks of the data density and typicality in the feature space [7], [9], [20], [24]. Fuzzy rules are then generated based on these identified prototypes. Due to the very high dimensionality of the feature vectors (4096 features) cosine dissimilarity is applied as the distance measure, as given by Equation (1) according to [7].

$$(d, x) = \sqrt{2 - 2\cos\theta_{x,y}} = \sqrt{2 - 2\frac{\sum_{i=1}^M x_i y_i}{\|x\|\,\|y\|}}$$
$$= \left\| \frac{x}{\|x\|} - \frac{y}{\|y\|} \right\|, \qquad (1)$$

where $\cos\theta_{x,y}$ is the angle between features vectors x and y; $\cos\theta_{x,y} = \frac{\langle x,y \rangle}{\|x\|\|y\|} = \frac{\sum_{i=1}^M x_i y_i}{\|x\|\|y\|}$; $\|x\| = \sqrt{\langle x,x \rangle} = \sqrt{\sum_{i=1}^M x_i^2}$ is the norm of x; M is the size of the vector (in this paper $M = 4096$). The cosine dissimilarity between the original vectors of the global features is equivalent to the Euclidean distance between the vectors normalized by their norms $(x \leftarrow \frac{x}{\|x\|})$, as described in Equation (1) [7]. This transformation is useful in order to reduce the computational complexity and it allows recursive calculation. So, as a default pre-processing step, all the extracted features are normalized. As aforementioned, the training process can be done for each fuzzy rule in parallel. Then, we can summarize the training process for the $c^{th} = (c = 1, 2, ...C)$ fuzzy rule as follows.

The cth fuzzy rule is initialized by the first image I_1^c of the corresponding class with feature vector denoted by $x_1^c = [x_{1,1}^c, x_{1,2}^c, ..., x_{1,M}^c]$. Meta-parameters of the DRB system are initialized as follows:

$$k \leftarrow 1; N^c \leftarrow 1; P_1^c \leftarrow I_1^c; \mu^c \leftarrow x_1^c; p^c \leftarrow x_1^c; S^c \leftarrow 1;$$
$$r_1^c \leftarrow r_0, \quad (2)$$

where k denotes a time instance; N^c is the number of prototypes observed; P_i^c is the first prototype of the fuzzy rule; μ^c is the global mean of all the feature vectors of the observed images of the cth class; p_1^c is the mean of feature vectors of the images associated with P_1^c, which is also the centre of the *data cloud* initialized by P_1^c; S_1^c is the support

of the first *data cloud*; r_1^c is the radius of the area of influence of the *data cloud*; r_0 is a small value used to stabilize the initial status of the newly identified prototypes.

In this paper, $r_0 = \sqrt{2 - 2\cos(\pi/6)} \approx 0.5176$; the rationale is that two vectors for which the angle between them is less than $\pi/6$ or 30° can be assumed to be pointing in close/similar directions. Therefore, we consider that r_0 is the measure that for two feature vectors with an angle between them smaller than 30 degrees can be considered to be similar. Note that r_0 is not a problem- or user- specific parameter. In fact, it can be defined without prior knowledge of the specific problem or data. One can note that the higher the value assumed for r_0, the fewer prototypes are identified during the training process. Consequently, more computationally efficient the DRB classifier will be. However, the DRB classifier may suffer from high generalization, not representing the real condition in detail.

With the first *data cloud*, the cth fuzzy rule is generated with the prototype P_1^c as follows:

R_1^c: IF $(I \sim P_1^c)$ THEN $(class\ c)$

Learning Algorithm:

As a new $(k+1)^{th}$ training image (I_k^c) that belongs to the cth class arrives, the global mean μ^c is firstly updated as follows:

$$\mu^c \leftarrow \frac{k-1}{k}\mu^c + \frac{1}{k}x_k^c. \qquad (3)$$

Data densities of $p_i^c(1, 2, ..., N^c)$ and feature vector of I_k^c, x_k^c are calculated according to [7].

$$D(z) = \frac{1}{1 + \frac{d^2(z,\mu^c)}{(\sigma^c)^2}}; z = p_1^c, p_2^c, ..., p_{N^c}^c, x_k^c, \qquad (4)$$

where, $\sigma^c = \sqrt{X^c - \|\mu^c\|^2} = \sqrt{1 - \|\mu^c\|^2}$; X^c is the average norm of feature vectors of the observed images, which is always equal to 1 due to the vector normalization operation. System structure and meta-parameters are updated to cope with the newly arrived image. The following principle is used in order to check whether I_k^c is a new prototype [25], [26].

$$IF\ (D(x_k^c) > \max_{j=1,2,...,N^c}(D(p_j^c)))$$
$$OR\ (D(x_k^c) < \min_{j=1,2,...,N^c}(D(p_j^c)))$$
$$THEN\ (I_k^c\ is\ a\ new\ prototype) \qquad (5)$$

If the condition is satisfied, I_k^c is set to be a new prototype and it initializes a new *data cloud*:

$$N^c \leftarrow N^c + 1; P_{N_c}^c \leftarrow I_k^c;$$
$$p_{N_c}^c \leftarrow x_k^c; S_{N_c}^c \leftarrow 1; r_{N_c}^c \leftarrow r_0. \qquad (6)$$

If the condition is not satisfied, then the algorithm searches for the nearest prototype to I_k^c, denoted by P_n^c, using the following equation:

$$P_n^c = \underset{j=1,2,...,N^c}{argmin} (d(p_j^c, x_k^c)). \tag{7}$$

Before associating I_k^c to the *data cloud* of P_n^c, a second condition is being checked in order to guarantee whether I_k^c is within the area influence of P_n^c:

$$IF \ (d(p_n^c, x_n^c) > r_n^c) \ THEN \ (I_k^c \ is \ a \ new \ prototype). \tag{8}$$

If the condition is met, it means that I_k^c is out of the influence area of P_n^c. Therefore, I_k^c becomes a new prototype of a new *data cloud* with meta-parameters initialized by Equation (2).

Otherwise, if the second condition is not met, I_k^c is assigned to the *data cloud* influenced by the prototype P_n^c and the parameters are updated as follows:

$$P_n^c \leftarrow \frac{S_n^c}{S_n^c + 1} P_n^c + \frac{1}{S_n^c + 1} x_k^c;$$
$$S_n^c \leftarrow S_n^c + 1;$$
$$r_n^c \leftarrow \sqrt{\frac{1}{2}(r_n^c)^2 + \frac{1}{2}(1 - \|p_n^c\|^2)}, \tag{9}$$

and the prototype P_n^c stay the same.

The Deep Rule-Based learning procedure algorithm is summarized below.

Deep Rule-Based Classifier: Learning Procedure

1: Read the first feature vector sample x_k^c of the image I_k^c;
2: Set $k \leftarrow 1; N^c \leftarrow 1; P_1^c \leftarrow I_1^c; \mu^c \leftarrow x_1^c; p^c \leftarrow x_1^c; S^c \leftarrow 1; r_1^c \leftarrow r_0$;
3: Create AnYa fuzzy rule: IF $(I \sim P_1^c)$ THEN $(class \ c)$;
4: **FOR** $k = 2, ...$
5:     Read $x_{(k)}^c$;
6:     Calculate μ^c according to Equation (3);
7:     Calculate $D(x_{(k)}^c)$ and $D(p_{(j)}^c)$ $(j = 1, 2, ..., N^c)$ according to Equation 4;
8:     **IF** Equation (5) holds
9:       Create rule according to Equation (6);
10:     **ELSE**
11:       Search for P_n^c according to Equation (7);
12:     **IF** Equation (8) holds
13:       Create rule according to Equation (6);
14:     **ELSE**
15:       Update rule according to Equation (9);
16:     **END**
17:    **END**
18: **END**

B. Semi-Supervised Learning

Unlabeled training images are available when the training process with labeled images finishes. Then, the DRB classifier can continue to learn from these unlabeled images. The unlabeled training images are defined as the set $\{U\}$, the number of unlabeled training images is defined as U.

The first step in the semi-supervised learning process, is to extract the vector of scores of confidence/degrees of closeness to the nearest prototypes for each unlabeled training image defined as $\lambda(U_i)$, $i = 1, 2, ..., U$. Scores of confidence are obtained according to the Equation (10).

$$\lambda^c = \underset{i=1,2,...,N^c}{max} (exp(-d^2(x, p_i^c))), \tag{10}$$

where λ^c denotes the confidence degree for c-th class. Labels are decided through the 'winners-take-all' principle, which is given by,

$$Label = \underset{c=1,2,...,C}{argmax} (\lambda^c(I)). \tag{11}$$

The recursive mean $\overline{\mu}_k$ of the λ^{max}, where $\lambda^{max}(U_i)$ is the highest score of confidence U_i obtained, is calculated according to the Equation (12) [26]. The recursive mean of the λ^{max} is used to detect sudden drop of the score of confidence generated by the DRB classifier when a new unknown class arrives.

$$\overline{\mu}_k = \frac{k-1}{k}\overline{\mu}_{k-1} + \frac{1}{k}\lambda_k^{max}, \overline{\mu} = \lambda_1^{max}. \tag{12}$$

Then the m-σ rule is applied. According to [27], the m-σ rule states that nearly all (99.73%) of values lie within 3 standard deviations of the mean if the data follows a Gaussian distribution, in more general cases a larger value, e.g. 6-σ is used. Therefore, if an unlabeled image U_i satisfies the following Equation (13) the DRB classifier is highly confident about the classes for the unlabeled training images. Then, these unlabeled images are used for updating the structure and meta-parameters of the DRB classifier.

$$IF \ (|\lambda^{max}(U_i) - \mu_k|) > m\sigma, \tag{13}$$

Otherwise, the DRB classifier is not highly confident about the class label, thus, these images are not used for updating the fuzzy rules structure. New classes are actively added as the proposed DRB classifier does not satisfies Equation (13). Rules are actively created according to the following Equation (14).

$$IF \ (|\lambda^{max}(U_i) - \mu_k|) < m\sigma$$
$$THEN \ (U_i \in (C+1)^{th} \ class). \tag{14}$$

Due to its prototype-based nature and its non-iterative training process, the ASSDRB classifier is more robust to incorrectly pseudo-labeled images. Incorrectly pseudo-labeled images can influence the ASSDRB classifier shifting the positions of some of the prototypes; and creating new false prototypes as wrong pseudo-labels are assigned. However, the ASSDRB classifier has a strong tolerance to a small amount of incorrectly pseudo-labeled images, because it is density-based.

III. Methodology

The Deep Rule-Based, K-nearest Neighbors (KNN) [28], Naive Bayes (NB) [29], Support Vector Machine (SVM) [30], and the deep convolutional neural network VGG–VD–16 classifiers are compared in terms of accuracy and processing time for labeled images. Moreover, the deep Rule-based method is used to actively classify unlabeled images.

The iROADS dataset [31] was considered in the analysis. The following dataset contains 4656 image frames recorded from moving vehicles on a diverse set of road scenes, recorded in day, night, under various weather and lighting conditions, as described below:

- Daylight - 903 images
- Night - 1050 images
- Rainy day - 1049 images
- Rainy night - 431 images
- Snowy - 569 images
- Sun strokes - 307 images
- Tunnel - 347 images

The iRoads dataset was divided into 90% for training and 10% for validation purposes. It is important to highlight that the 'Sun stroke' and 'Tunnel' classes were left from the training phase in order to be presented as unlabeled images for the classifiers. Classification accuracy is calculated as follows:

$$ACC(\%) = \frac{TP + TN}{TP + FP + TN + FN}, \tag{15}$$

where TP, FP, TN, FN denote true and false, negative and positive, respectively.

All the experiments were conducted with MATLAB 2018a using a personal computer with a 1.8 GHz Intel Core i5 processor, 8-GB RAM, and MacOS operating system. The classification experiments were executed using 10-fold cross validation under the same ratio of training-to-testing sample sets. Default parameters were chosen for KNN [32], NB [8], SVM [32], and VGG–VD–16 [33].

## IV. Classification Results

A. labeled Images

Computational simulations were performed to assess the accuracy of the classification methods considering the labeled images. Table I summarizes the results obtained by the DRB, KNN, NB, SVM and VGG–VD–16 for the iRoads dataset considering the classes "Daylight, Night, Rainy Day, Snowy".

TABLE I
PERFORMANCE COMPARASION: LABELED IMAGES

Method	*Accuracy*	Time(s)	# Prototypes
VGG–VD–16	**99.51 %**	836.28	–
DRB	99.02%	**2.95**	**521**
SVM	94.17%	5.67	–
KNN	93.49%	4.43	–
NB	88.35%	5.31	–

As expected, Table I shows that the DCNN VGG–VD–16 approach has the higher accuracy performance. However,

one can note that the DRB model, using 0.35% of the time required by VGG-VD-16 and much less computational resources, could reach 99% classification rate. Moreover, due to its rule-based nature, the DRB model produced 521 prototypes. Since each prototype is a 4096-dimensional vector if we consider 4-byte (32-bit) each word to represent a real value we get approximately 8MB memory needed to store all prototypes. These are human interpretable, as the following fuzzy rule:

IF (I ∼ 〔image〕) OR

(I ∼ 〔image〕) OR ...

OR (I ∼ 〔image〕)

THEN 'Daylight scene'
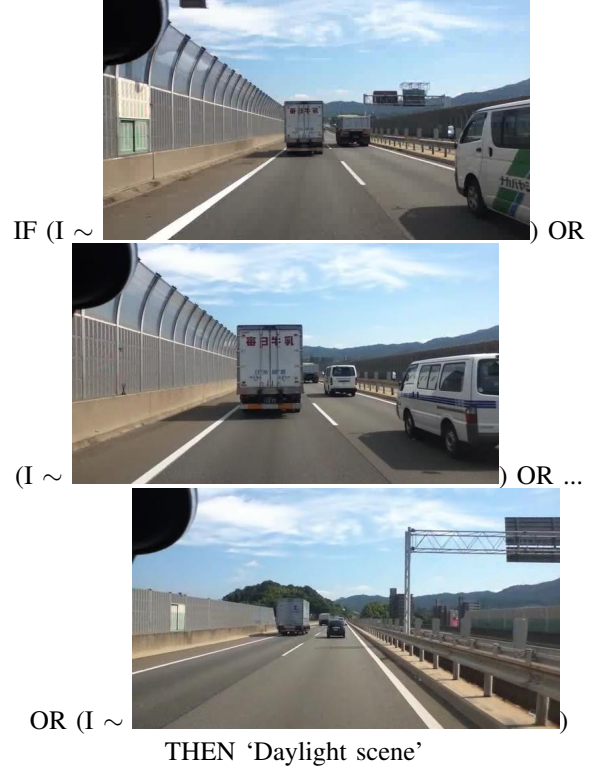
AnYa type fuzzy rules generated by the DRB model provide a very intuitive representation for humans. Moreover, each of the AnYa type fuzzy rules can be interpreted as a number of simpler fuzzy rules with single prototype connected by 'OR' operator. As a result, a massive parallelization is possible. The transparent process provided by the DRB model favors understandability of the system, differing from the DCNN approach, which is called 'black box', since it hides (due to its nature) from users all the insights used to generate the final resulting structure.

B. Unlabeled Images

1) Tunnel Scenario: Results for unlabeled images are shown in this subsection. The first step was to classify a 'Tunnel' road scene with the pre-trained VGG-VD-16. As the DCNN VGG-VD-16 structure was not trained for this type of situation, it classified the 'Tunnel' scenario as a 'Sun stroke' scene with 74% of accuracy, as shown in Figure 2. The misclassification in this case could be a potential risk for a self-driving car, as in a 'Sun stroke' situation the driving conditions are different than a 'Tunnel' condition.
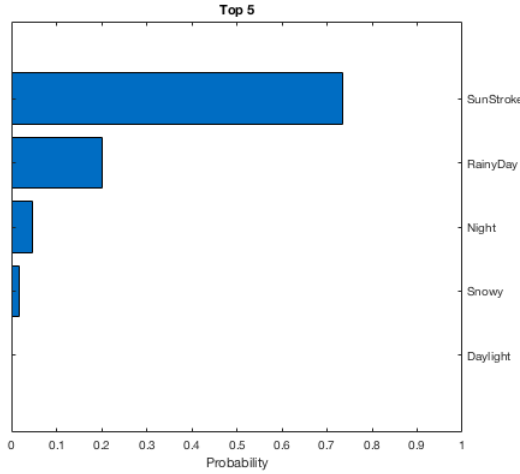
Fig. 2. VGG-VD-16 - TOP 5 classification for Tunnel driving scene

Therefore, the next step was to apply the proposed Actively Semi-Supervised Deep Rule-based classifier (ASSDRB) in order to classify and validate the new arriving scenario.
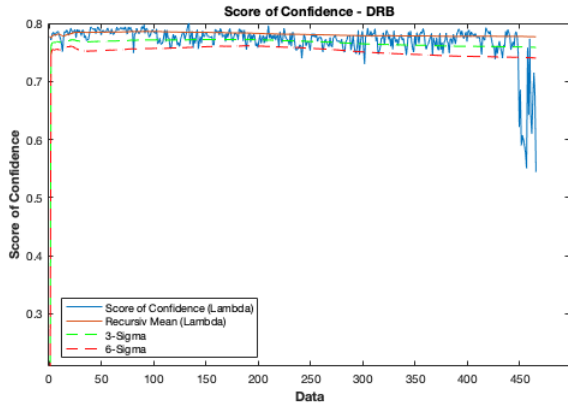

Fig. 3. Score of confidence for Tunnel driving scene

From Figure 3, one can see that the score of confidence given by the DRB model drastically fall from the image ID 449 to 450. This is precisely the moment where there is a shift in the concept and the lighting condition changes. Therefore, the DRB classifier model is able to detect this concept change due to its prototype and density-based nature. Note that the blue line given in the Figure 3 is the maximum value of λ, used by the DRB model according to the 'winner-takes-all' system to assign class labels to the prototypes, the red line is the recursively calculated mean of the maximum value of λ. Through the use of a recursive mean of the maximum value of λ it is possible to detect the drastic fall in the score of confidence when a new scene is shown. Therefore, it is very clear that the prototype belongs to a new class, as the AnYa fuzzy rule is autonomously created:

Therefore, after the proposed ASSDRB classifier has been primed with labeled training images, it is run to learn 'on-the-fly' from the images in a semi-supervised manner [20]. Table

R_{new}: IF (I $\sim$)
THEN 'New class 1'

II presents the classification result autonomously obtained by the ASSDRB for the proposed problem.

TABLE II
PERFORMANCE COMPARASION: UNLABELED IMAGES ("TUNNEL SCENARIO")

m-σ	$Accuracy$	Time(s)	# Prototypes
3-σ	95.45 %	57.63	550
6-σ	**98.72 %**	52.25	543

As we can note from Table II, the ASSDRB could obtain accuracy of 98.72% in a fully unsupervised manner using 6-σ. This was possible due to recursive calculation of the mean maximum value of λ and the prototype nature of the method which allowed to detect the fall in the given score of confidence assign the new unseen scene to a new class autonomously. One AnYa fuzzy rule (with prototypes linked with logical OR connective) were autonomously generated in the primer structure. It is possible to note that 3-σ is not efficient in terms of accuracy as 6-σ because the data does not follow Gaussian distribution and then 6-σ works better in this case.

2) Rainy Day Scenario: As in the previous simulation, the DLNN VGG–VD–16 was trained omitting the 'Rainy day' scenario. After the training phase, the 'Rainy day' scene was presented to the deep neural network approach. As the DLNN approach was not trained for the presented situation, it wrongly classified the scene, as shown in Figure 4. The DLNN approach classified the 'Rainy day' scenario with almost 90% of confidence as a 'Night' scenario. In a real situation, a misclassification as the one presented, could be a potential risk of accident for autonomous cars.

The unlabeled images were then presented to the proposed ASSDRB classifier. Then, the newly density-based method for automatic detection of unseen scenes could detect the fall in the score of confidence for this classification task. As one can observe through the recursive mean $\overline{\mu}_k$ of the λ^{max}, even though the proposed classifier was not trained for this lighting condition, it could detect a notable fall in the score of confidence (λ^{max}) significantly below 3σ, indicating that a new unknown class is presented to the classifier. Figure 5 illustrates the results of obtained in terms of score of confidence (λ^{max}) and recursive mean $\overline{\mu}_k$ of the λ^{max} for the proposed test.

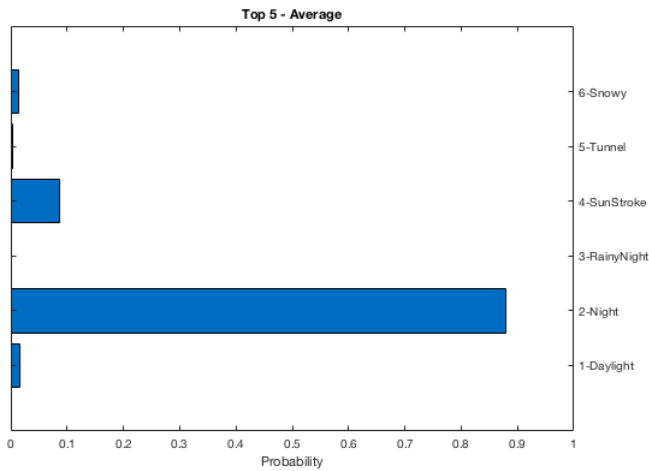Therefore, using the ASSDRB model a new class could be created actively as shown by the AnYa rule below.

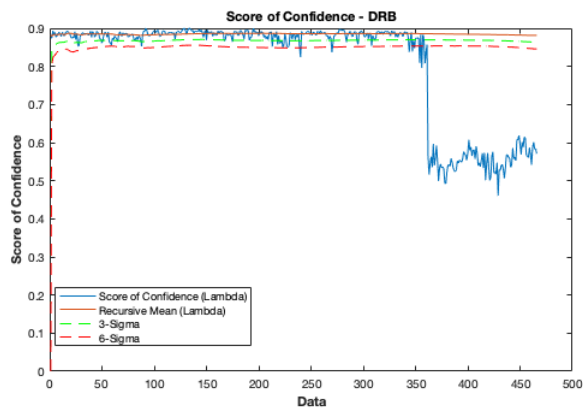Fig. 4. VGG-VD-16 - TOP 5 classification for Rainy Day driving scene

Fig. 5. Score of confidence - DRB

R_{new}: IF (I ∼)

THEN 'New class 2'

TABLE III
PERFORMANCE COMPARASION: UNLABELED IMAGES ('RAINY DAY SCENARIO')

m-σ	$Accuracy$	Time(s)	# Prototypes
3-σ	97.32 %	48.51	521
6-σ	**99.12 %**	45.43	517

Classification results obtained using unlabeled 'Rainy Day' scenarios are shown in Table III. The proposed ASSDRB model fully autonomously obtained 99.12% of accuracy in the best scenario. As the shift in the scene condition is very

clear, the deep rule-based model was able to detect a big fall in the score of confidence; therefore, it was highly confident that the new unseen scenes presented belonged to a new class. DRB is evidently fast in all cases. It is notable that 6-σ provides better results due to the data distribution. Moreover, it presents a more compact model, as less prototypes are necessary to build the rules.

In general, experiments on different lighting conditions for driving have shown that the deep neuro-fuzzy modeling is an efficient framework for classification tasks. Classification accuracies were higher than those produced by the traditional statistical methods, such as KNN, NB and SVM. The number of algebraic calculations for mainstream Deep Neural Networks are larger and, therefore, the method is much slower than DRB classifier. Moreover, DRB produced transparent linguistic fuzzy rules, which are human interpretable.

Usually, time to process data and adapt a fuzzy model is not a constraint on image-classification problems, but it may be an issue in higher-frequency data stream applications in real-time. DRB adaptation deals with nonstationarities very efficiently and fast. Naturally, DRB becomes interesting for real-time classification scenarios.

V. CONCLUSION

In this paper, we propose an active semi-supervised deep rule-based (ASSDRB) classifier in order to classify unseen roads scenes actively. Experiments have shown that the proposed deep neuro-fuzzy rule-based classifier is efficient considering the classification accuracy and data processing speed. Moreover, it has the ability to learn 'on-the-fly', as it is equipped with an incremental learning engine. The main novelties are:

- We offer a new mechanism to automatically and reliably detect new unseen scenes based on confidence drop.
- We apply a new, recently developed DRB classifier to driverless cars scenario for the first time and compare with mainstream alternatives.

In general, the proposed deep neuro-fuzzy rule-based classifier has demonstrated to be able to self-adapt its structure and provide human understandable *IF ... THEN* fuzzy rule-based system structure. Its prototype-based nature provides a way to deal with large volumes of data. Also the approach is an effective and efficient way of handling data streams due to its ability to adapt models to different situations and provide quick response to changes, i.e. to concept drifts and shifts.

Future research will concentrate on the development of recursive equations based on information specificity, correntropy and cardinality to guide the deep neuro-fuzzy model adaptation. Also, hierarchical multi-layer architecture will be studied. Moreover, video simulation will be considered in order to replicate a realistic scenario.

REFERENCES

[1] P. P. Angelov, X. Gu, and J. C. Príncipe, "Autonomous learning multimodel systems from data streams," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2213–2224, 2018.

[2] E. A. Soares, H. A. Camargo, S. J. Camargo, and D. F. Leite, "Incremental gaussian granular fuzzy modeling applied to hurricane track forecasting," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

[3] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 3, pp. 362–370, 2018.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[6] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.

[7] P. P. Angelov and X. Gu, *Empirical approach to machine learning*. Springer, 2018, isbn: 978-3-030-02384-3.

[8] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[9] P. P. Angelov and X. Gu, "Deep rule-based classifier with human-level performance and characteristics," *Information Sciences*, vol. 463–464, pp. 196–213, 2018.

[10] E. Soares, P. Costa Jr, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Applied Soft Computing*, vol. 64, pp. 445–453, 2018.

[11] D. Leite, R. Ballini, P. Costa, and F. Gomide, "Evolving fuzzy granular modeling from nonstationary fuzzy data streams," *Evolving Systems*, vol. 3, no. 2, pp. 65–79, 2012.

[12] P. Angelov, D. P. Filev, and N. Kasabov, *Evolving intelligent systems: methodology and applications*. John Wiley & Sons, 2010, vol. 12.

[13] E. Lughofer, M. Pratama, and I. Skrjanc, "Incremental rule splitting in generalized evolving fuzzy systems for autonomous drift compensation," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 1854–1865, 2018.

[14] N. K. Kasabov and Q. Song, "Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[15] P. Angelov and N. Kasabov, "Evolving intelligent systems, eis," *IEEE SMC eNewsLetter*, vol. 15, pp. 1–13, 2006.

[16] E. Lughofer and P. Angelov, "Handling drifts and shifts in on-line data streams with evolving fuzzy systems," *Applied Soft Computing*, vol. 11, no. 2, pp. 2057–2068, 2011.

[17] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1025–1032.

[18] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[19] J. Kim and J. F. Canny, "Interpretable learning for self-driving cars by visualizing causal attention." in *ICCV*, 2017, pp. 2961–2969.

[20] X. Gu and P. P. Angelov, "Semi-supervised deep rule-based approach for image classification," *Applied Soft Computing*, vol. 68, pp. 53–68, 2018.

[21] P. Angelov and R. Yager, "Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density," in *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 2011, pp. 62–69.

[22] Y. Liu, Y. Liu, and L. Ding, "Scene classification based on two-stage deep feature fusion," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 183–186, 2018.

[23] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *International Journal of General Systems*, vol. 41, no. 2, pp. 163–185, 2012.

[24] X. Gu, P. P. Angelov, and J. C. Príncipe, "A method for autonomous data partitioning," *Information Sciences*, 2018.

[25] P. P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1462–1475, 2008.

[26] P. Angelov, *Autonomous learning systems: from data streams to knowledge in real-time*. John Wiley & Sons, 2012.

[27] P. J. Rousseeuw and M. Hubert, "Robust statistics for outlier detection," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 73–79, 2011.

[28] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," *Multiple Classifier Systems*, vol. 34, no. 8, pp. 1–17, 2007.

[29] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. IBM New York, 2001, pp. 41–46.

[30] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[31] M. Rezaei and M. Terauchi, "Vehicle detection based on multi-feature clues and dempster-shafer fusion theory," in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2013, pp. 60–72.

[32] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[33] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 689–692.