Received April 5, 2022, accepted April 29, 2022, date of publication May 16, 2022, date of current version May 27, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3175195

Reducing Overconfidence Predictions in Autonomous Driving Perception

GLEDSON MELOTTI[1,2], CRISTIANO PREMEBIDA[2], JORDAN J. BIRD[3], DIEGO R. FARIA[4], AND NUNO GONÇALVES[2,5], (Member, IEEE)

[1]Federal Institute of Espirito Santo, Espírito Santo 29932-540, Brazil
[2]Department of Electrical and Computer Engineering, Institute of Systems and Robotics (ISR-UC), University of Coimbra, 3030-290 Coimbra, Portugal
[3]Computational Intelligence and Applications Research Group (CIA), Department of Computer Science, Nottingham Trent University, Nottingham NG1 4FQ, U.K.
[4]School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, Hertfordshire AL10 9AB, U.K.
[5]Portuguese Mint and Official Printing Office, 1000-042 Lisbon, Portugal

Corresponding author: Gledson Melotti (gledson@ifes.edu.br)

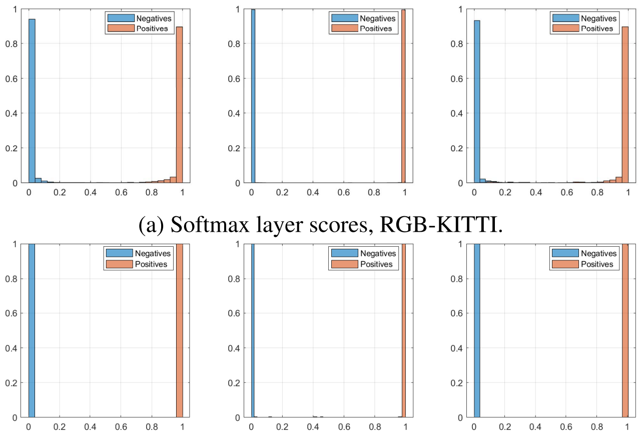
ABSTRACT In state-of-the-art deep learning for object recognition, Softmax and Sigmoid layers are most commonly employed as the predictor outputs. Such layers often produce overconfidence predictions rather than proper probabilistic scores, which can thus harm the decision-making of 'critical' perception systems applied in autonomous driving and robotics. Given this, we propose a probabilistic approach based on distributions calculated out of the Logit layer scores of pre-trained networks which are then used to constitute new decision layers based on Maximum Likelihood (*ML*) and Maximum a-Posteriori (*MAP*) inference. We demonstrate that the hereafter called *ML* and *MAP* layers are more suitable for probabilistic interpretations than Softmax and Sigmoid-based predictions for object recognition. We explore distinct sensor modalities via RGB images and LiDARs (RV: range-view) data from the KITTI and Lyft Level-5 datasets, where our approach shows promising performance compared to the usual Softmax and Sigmoid layers, with the benefit of enabling interpretable probabilistic predictions. Another advantage of the approach introduced in this paper is that the so-called *ML* and *MAP* layers can be implemented in existing trained networks, that is, the approach benefits from the output of the Logit layer of pre-trained networks. Thus, there is no need to carry out a new training phase since the *ML* and *MAP* layers are used in the test/prediction phase. The Classification results are presented using reliability diagrams, while detection results are illustrated using precision-recall curves.

INDEX TERMS Bayesian inference, confidence calibration, object recognition, perception system, probability prediction.

I. INTRODUCTION

Recent advances in deep learning and sensory technology (*e.g.*, RGB cameras, LiDAR, radar, stereo, RGB-D, among others [1], [2]) have made remarkable contributions to perception systems applied to autonomous driving [3]–[6]. Perception systems include, but are not limited to, image and point cloud-based classification and detection [5], [7]–[10], semantic segmentation [3], [11], [12], and tracking [13], [14]. Oftentimes, regardless of the type of network architecture or input modalities, most state-of-the-art CNN-based object recognition algorithms output normalized prediction scores via the Softmax layer [15] *i.e.*, the prediction values are in a range of [0, 1], as shown in Fig. 1. Furthermore, such algorithms are often implemented through deterministic neural networks, and the prediction itself does not consider the model's actual confidence for the predicted class in decision-making [16]. In fact, in most cases, the decision-making takes into account only the prediction value provided directly by a deep learning algorithm disregarding a proper level of confidence of the prediction (which is unavailable for most networks). Therefore, evaluating the prediction confidence or uncertainty is crucial in decision-making because an erroneous decision can lead to disaster, especially in autonomous driving where the safety of human lives are dependent on the automation algorithms.

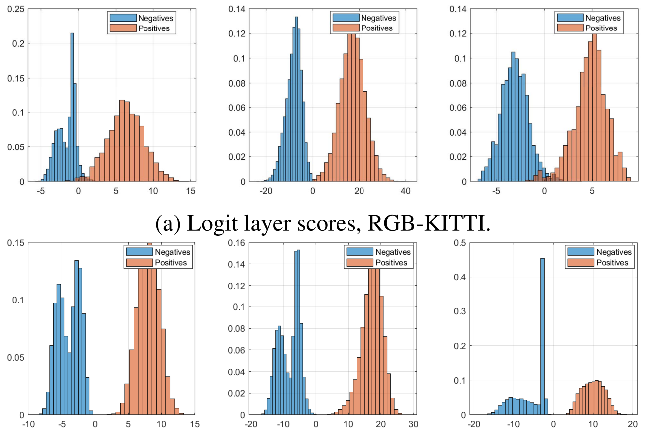
The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.



(a) Softmax layer scores, RGB-KITTI.

(b) Softmax layer scores, RGB-LL5.

FIGURE 1. Graphs (a) and (b) are the Softmax prediction scores for the 'pedestrian', 'car' and 'cyclist' classes (where the positives are in orange), showing evidence of overconfidence behavior. The bar-plots were obtained on a RGB image classification set from the KITTI and LL5 databases respectively.



(a) Logit layer scores, RGB-KITTI.

(b) Logit layer scores, RGB-LL5.

FIGURE 2. Probability density functions (PDFs), using normalized histograms, for the Logit layers data on the training sets of the KITTI (a) and LL5 (b) datasets. The graphs are organized from left-right by classes (pedestrian, car and cyclist, where the positives are in orange) using the RGB modality.

Many works have pointed out Softmax layer overconfidence as an open issue in the field of deep learning [17]–[20]. Two main techniques have been suggested to mitigate the overconfidence in deep networks, calibration [21]–[27] and regularization [24], [25], [28]. Often, calibrations are defined as techniques that act directly on the resulting output of the network, while regularization are techniques that aims to penalize network weights through a variety of methods, which adds parameters or terms directly to the network cost/loss function [28]–[30]. However, the paper proposed by [31] defines regularization techniques as a type of calibration. Consequently, the latter demands that the network must be retrained.

The overconfidence problem is more evident in complex networks such as Convolutional Neural Networks (CNNs), particularly when using the Softmax layer as the prediction layer, thus generating ill-distributed outputs *i.e.*, values close to either zero or one [23] which can be observed in Fig. 1a and Fig. 1b. We note that this is desirable when the true positives have higher scores. However, the counterpart problem is that 'overconfidence networks' also generate high-score values for the objects erroneously detected or classified *i.e.*, false positives. Given this problem, a question that arises, *how can we guarantee prediction values that are 'high' for true positives and, at the same time, 'low' for false positives?* This question can be answered by analyzing the output of the network's Logit layer, which provides a smoother output than the Softmax layer. This can be observed within Figs. 2a and 2b.

Following this, we can put a new question: *although normalized outputs aim to guarantee a 'probabilistic interpretation', how reliable are these predictions? Additionally, given an object belonging to a non-trained/unseen class (e.g., an unexpected object on the road), how confident is the model's prediction?* These are the key research questions explored in this work by considering the importance of having models grounded on interpretable probability assumptions to enable adequate interpretation of the outputs, ultimately leading to more reliable predictions and decisions. In terms of contributions, this paper introduces new prediction layers, designated Maximum Likelihood (*ML*) and Maximum a-Posteriori (*MAP*) layers, for deep neural networks, which provide a more adequate solution compared to state-of-the-art (Softmax or Sigmoid) prediction layers. Both *ML* and *MAP* layers compute a single estimate, rather than a distribution. Moreover, this work contributes towards the advances of multi-sensor perception (RGB and LiDAR modalities) for autonomous perception systems [32]–[34] by proposing a probability-grounded solution that is practical in the sense it can be used in existing (*i.e.*, pre-trained) state-of-the-art models such as Yolo [35].

It is important to emphasize that there is no need to retrain the neural networks when the approach described in this article is employed, because the *ML* and *MAP* prediction layers produce outputs based on PDFs obtained from the Logits of already trained networks. Therefore, instead of using the traditional prediction layers (Softmax or Sigmoid) to predict the object scores on a test set, the *ML* and *MAP* nonlinearities can be used to make the predictions for the objects scores. Thus, the proposed technique in this paper is practical given that a network has already been trained with Softmax (*SM*) or Sigmoid (*SG*) prediction layers. In other words, the *ML* and *MAP* layers depend on the Logit's outputs of the already trained network[1]

[1] A note for the reviewers: this paper is an extension of our workshop-paper [36], as well as an extension of the paper [37]. The main difference between this paper and the two previously mentioned papers is in the analysis of the results through reliability diagrams, considering the expected calibration error, and maximum calibration error metrics. In addition, this paper considers a more detailed analysis regarding the predicted score values on out-of-training distribution test data (unseen class).

In summary, the scientific contributions arising from this work are:

- An investigation of the distribution of predicted values of the Logit and Softmax layers, for both calibrated and non-calibrated networks;
- An analysis of the predicted probabilities inferred by the proposed *ML* and *MAP* formulations, both for object classification and detection;
- An investigation of the predicted score values on out-of-training distribution test data (unseen/non-trained class);
- The proposed approach does not require the retraining of networks;
- Experimental validation of the proposed methodology through different modalities, RGB and Range-View (3D point clouds-LiDAR), for classification (using InceptionV3) and object detection (using YoloV4).

In this paper, we report on object recognition results showing that the Softmax and Sigmoid prediction layers do indeed sometimes induce erroneous decision-making, which can be critical in autonomous driving. This is particularly evident when 'unseen' samples *i.e.*, out-of-training distribution test data are presented to the network. On the other hand, the approach described here is able to mitigate such problems during the testing stage (prediction).

The rest of this article is structured as follows. The related work is presented in Section II, while the proposed methodology is developed in Section III. The experimental part and the results are reported in Section IV, the conclusion is given in Section V, while Section VI presents ideas to expand the proposed research, and finally Section VI (Appendix) presents results considering an extra experiment.

II. RELATED WORK

In this section, we review the key methodologies related to our proposed approach. We briefly discuss the uncertainties of neural networks based on the concepts of Bayesian inference, consequently defining the types of uncertainties that can be captured by the Bayesian Neural Networks (BNNs). Then, techniques for reducing overconfidence of prediction layers are presented as well, in particular the regularization and calibration techniques.

A. PREDICTIVE UNCERTAINTY

Many deep learning methods used for perception systems (objects detection and recognition) do not capture the network uncertainties at training and test times. The Bayesian Neural Network (BNN) is an alternative to cope with uncertainties and it can be carried out through distinct approaches. One way is to obtain the posterior distribution using variational inference after defining a prior distribution to the network weights [29], [38], [39]. Another method is the ensemble of multiple networks with the same architecture and different training sets for estimating predictive uncertainty [40].

Currently, many studies consider aleatory and epistemic uncertainties obtained through BNNs. Aleatory uncertainty is related to the inherent noise of observations (uncertainties arising from sensor inherent noise and associated with the distance of the object to be detected, as well as the object occlusion), while the epistemic ones explain the uncertainties in the model parameters (uncertainties of the model associated with the detection accuracy, showing the limitations of the model) [41]. The formulation of aleatory and epistemic uncertainties with the aim of presenting confidence of predictions, which can capture the uncertainties in object recognition, can be done through BNNs, Shannon Entropy (uncertainty in the prediction output) and Mutual Information (confidence of the model in the output) to measure the uncertainty of the classification scores [42]–[44].

The uncertainty of a prediction can also be achieved through Monte Carlo dropout strategy, using the dropout layers at test time *i.e.*, the predicted values depend on the randomly chosen connections between the neurons according to the dropout rate, that is, the same test example (an object) forwarded several times in the network can have different predicted values (the predicted values are not deterministic). In this way, it is possible to obtain the distribution, the average (final predicted value) and the variance (uncertainty) [45] for each example.

Differing from the aforementioned works, the approach proposed in this paper uses data obtained from the Logit layer of already trained/existing networks, to employ the concepts of Bayesian inference. The methodology proposed in this paper defines a final prediction value for each object and does not need to predict recurrently for the same object several times. Furthermore, the approach presented in the paper does not consider the distribution of the network weights, and thus, it is an efficient and practical approach. These advantages are clear when compared to traditional Bayesian neural networks and the Monte Carlo dropout strategy, because the novel strategy presented here avoids a high computational cost and at the same time does preserve the recognition/detection performance. Nevertheless, there are ongoing research on Bayesian neural networks that have reduced the computational cost through feature decomposition and memorization [46].

B. REGULARIZATION AND CALIBRATION

Another important component for the improvement of the predicted values are the regularization techniques that avoid overfitting and contribute to reduce overconfidence predictions, such as the transformation of network weights using $L1$ and $L2$ [47] regularization, label and model regularization by a process of pseudo-label and self-training [30], label smoothing [48], knowledge distillation [49], architecture development where the network has to determine whether or not an example belongs to the training set, and specific cost mathematical formulation [50], [51]. Other well-known regularization techniques are the Batch Normalization [52], stochastic regularization techniques such as Dropout [53], multiplicative Gaussian noise [54], and dropConnect [55].

Alternatively, highly confident predictions can often be mitigated by calibration techniques such as temperature scaling (*TS*) [23], by multiplying all the values of the logit vector by a scalar parameter, $\frac{1}{TS} > 0$, for all classes, where the value of *TS* is obtained by minimizing the negative log likelihood on the validation set; Isotonic Regression [56] which combines binary probability estimates of multiple classes, thus jointly optimizing the bin boundary and bin predictions; Platt Scaling [57] which uses classifier predictions as features for a logistic regression model; Beta Calibration [58] which uses a parametric formulation that considers the Beta probability density function; compositional method (parametric and non-parametric approaches) [59], as well as the embedding complementary networks technique [60], [61].

In this study, we reduce highly confident predictions on the test set by replacing the predicted values by Softmax and Sigmoid layers with the predicted values from *ML* and *MAP* nonlinearities, obtaining a smoother score distribution for new objects. Such functions depend on the output of the network's Logit layer, by means of parametric (Gaussian functions) and nonparametric (normalized histograms) modeling. This is a post-training operation, that is, the novel inference functions proposed in this work do not modify the weights neither the cost function of the network and still provides very satisfactory results. This is an advantage over regularization techniques, since the *ML* and *MAP* layers do not require network retraining. The advantage of the approach proposed in this paper with respect to calibration techniques is to provide a smoother distribution of the predicted values without degrading the results.

III. PROPOSED METHOD

This section presents the core of the proposed methodology *i.e.*, the formulations for making predictions based on the novel *ML* and *MAP* prediction layers. The development of such a methodology begins with the concepts of probabilities, random variables, distribution function, probability density function and Bayes' theorem *i.e.*, the background to develop the methodology proposed in this paper. In the second stage, we present the proposed method through formulations of the Maximum Likelihood (*ML*) and Maximum a-Posteriori (*MAP*) layers, as well as nonparametric and parametric mathematical modeling to define the posterior (likelihood-conditional) and prior probabilities. Finally, we present the network architectures, diagrams for evaluating the calibration of the proposed methodology, and the datasets that have been used in the experiments.

A. A BRIEF REVIEW OF PROBABILITY AND DENSITY FUNCTIONS

The output scores $\mathbf{x} = \{x_1, \ldots, x_{nc}\}$ of a supervised classification system with nc classes, $\mathbf{c} = \{c_1, \ldots, c_{nc}\}$ can be formulated according to a random experiment considering a sample space $\mathbf{S}$. The numerical outcome obtained from each element of $\mathbf{S}$ is related to a real number defined by the random variable (rv) $\mathbf{x}$ *i.e.*, the output scores, which is conditioned to the rv c. Formally, the rv is a function that maps each element of the sample space with a real number of the set $\mathbb{R}$, which can be simply expressed as $\mathbf{x} : \mathbf{S} \rightarrow \mathbb{R}$. In other words, an rv is a function $\mathbf{x}$ that outputs a real number $\mathbf{x}(\zeta)$ for each element $\zeta \in \mathbf{S}$ of a random experiment. From the sample space, an event (subset of $\mathbf{S}$) can be defined and associated with a probability $\mathbf{P}$ between the $[\xi, \xi + \Delta\xi]$ interval. Such probability is a distribution function and its derivative is the probability density function (PDF) $f_x(x = \xi)$, as in (1) [62].

$$f_x(x = \xi) = \lim_{\Delta\xi \to 0} \frac{\mathbf{P}\{\xi \leq \mathbf{x} \leq \xi + \Delta\xi\}}{\Delta\xi}, \tag{1}$$

where $f_x(x = \xi) \geq 0 \; \forall \; \xi$, considering ξ continuous. The integral of (1) represents the probability $\mathbf{P}$ with the random variable $\mathbf{x}$ contained in the interval. Consequently, if the interval $[\xi, \xi + \Delta\xi]$ is sufficiently small, the probability will be $\mathbf{P}\{\xi \leq \mathbf{x} \leq \xi + \Delta\xi\} \simeq f_x(x = \xi)\Delta\xi$ *i.e.*, the probability of the random variable $\mathbf{x}$ is proportional to $f_x(x = \xi)$. Thus, the probability will be maximum if the interval $[\xi, \xi + \Delta\xi]$ contains its value and $f_x(x = \xi)$ will be maximum. Such a value is the most likely value of $\mathbf{x}$.

Given the most likely value of the random variable $\mathbf{x}$, Maximum Likelihood (*ML*) and Maximum a-Posteriori (*MAP*) inferences can be obtained. However, the random variable $\mathbf{x}$ is dependent of the variable $\mathbf{c}$ for the formulation of *ML* and *MAP*. Therefore, the density function is conditional to $\mathbf{c}$ [62], as formulated in (2):

$$f_x(x = \xi|\mathbf{c}) = \lim_{\Delta\xi \to 0} \frac{\mathbf{P}\{\xi \leq \mathbf{x} \leq \xi + \Delta\xi|\mathbf{c}\}}{\Delta\xi}. \tag{2}$$

If the random variable is discrete, a probability mass function (PMF) is used instead of a probability density function (PDF). Assuming that the class conditional probability $P(\mathbf{x}|\mathbf{c})$ (likelihood) and the prior are known, the posterior probability $P(\mathbf{c}|\mathbf{x})$ can be obtained through Bayes' rule

$$P(\mathbf{c}|\mathbf{x}) = \frac{P(\mathbf{x}|\mathbf{c})P(\mathbf{c})}{P(\mathbf{x})}, \tag{3}$$

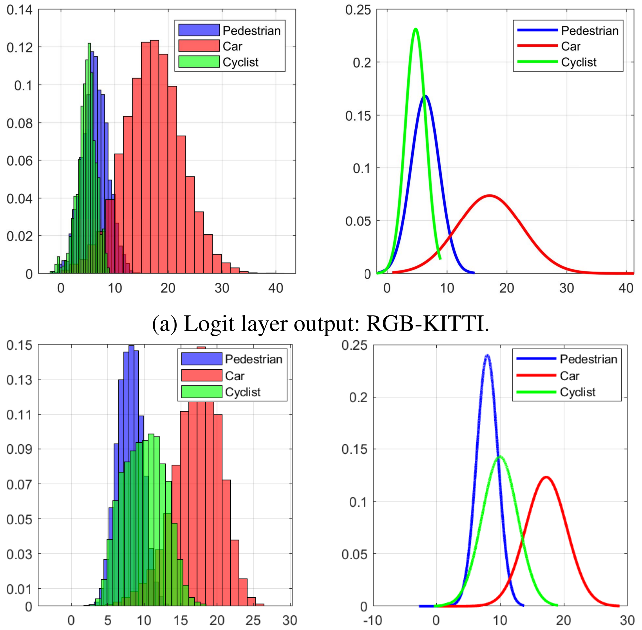
where $P(\mathbf{c})$ is the prior probability, $P(\mathbf{x}) \neq 0$ is the marginal probability defined by $\int P(\mathbf{x}|\mathbf{c})(\mathbf{c})dc$, that often can be determined by law of the total probability [63]. Thus, (3) can be re-written using the *per-class* expression:

$$P(c_i|\mathbf{x}) = \frac{P(\mathbf{x}|c_i)P(c_i)}{\sum_{i=1}^{nc} P(\mathbf{x}|c_i)P(c_i)}. \tag{4}$$

In this work, the goal is to use (4) to make inferences on the test set about the 'unknown' rv $\mathbf{c}$ from the dependence with $\mathbf{x}$ *i.e.*, the value of the posterior distribution of $\mathbf{c}$ is determined after observing the value of $\mathbf{x}$.

B. ML AND MAP PREDICTION LAYERS

The proposed *ML* and *MAP* layers make inference based on PDFs obtained from the Logit layer prediction scores by using the training set. This is illustrated in Fig. 3, where the horizontal axes represent the random variable $\mathbf{x}$ and the vertical axes are the normalized frequency of



(a) Logit layer output: RGB-KITTI.

(b) Logit layer output: RGB-LL5.

FIGURE 3. **From left-right respectively, normalized histogram-based densities and Gaussian densities calculated on the Logit layer values, for each class, on the training set (here for the RGB modality). On the 1st row, we have the densities on the KITTI set while the 2nd row shows the densities on the LL5 training set.**

the amount of objects in the classification and detection datasets. We can observed that the distribution scores from the Logit layer are far more appropriate to represent a PDF (as shown in Fig. 2). Therefore, the *ML* and *MAP* layers are more adequate to perform probabilistic inference in regard to permitting decision-making under uncertainty, which is particularly relevant in autonomous driving and robotic perception systems.

As noted in (4), the posterior probability depends on the class conditional probability (likelihood function) and on the prior probability *i.e.*, the *MAP* estimated depends on a distribution for both the likelihood and prior, while *ML* only depends on $P(\mathbf{x}|\mathbf{c})$, because $P(\mathbf{c})$ is usually assumed to be uniform and identically distributed. The probabilities $P(\mathbf{x}|\mathbf{c})$ are modeled by means of non-parametric estimates over the predicted scores of the Logit layer for each class, as showed in the first column of Fig. 3. These estimates are obtained on the training set, through normalized histograms (*i.e.*, discrete densities defined by a single parameter - the number of bins) for each modality, as shown in the Table 1.

Histograms are graphical ways of summarizing or describing a variable in a simple way, in other words, histograms show how variables (in this case, the network logits) are distributed, revealing modes and bumps, as well as information about the frequencies of observations. As said by C. Bishop [63], 'we can view the histogram as a simple way to model a probability distribution given only a finite number of points drawn from that distribution'. Often, the bins of a histogram are chosen to have the same width

TABLE 1. **Number of bins and smoothing parameter (λ) for *ML* and *MAP* layers.**

		Maximum Likelihood		
	RGB Modalitiy		RV Modality	
Dataset	Bins	Additive Smoothing	Bins	Additive Smoothing
KITTI	25	1.0×10^{-2}	25	1.0×10^{-2}
LL5	25	1.0×10^{-2}	30	1.0×10^{-2}
		Maximum a-Posteriori		
	RGB Modalitiy		RV Modality	
Dataset	Bins	Additive Smoothing	Bins	Additive Smoothing
KITTI	25	1.0×10^{-2}	25	1.0×10^{-2}
LL5	25	1.0×10^{-2}	30	1.0×10^{-2}

thus, the only (single) parameter left is the number of bins (nbins). To do so, nbins can be mathematically determined by means of the mean squared error (MSE-expected value of the squared error) [64]. However, for our methodology, we have chosen nbins empirically to guarantee a result very close to or better than the results provided by the *SM* and *SG* layers and, in addition, to generate smoother distribution by adding the parameter λ. Thus, the process of estimating the number of bins and λ (the additive smoothing factor) have been defined empirically by verifying which combinations would not degrade the results. So, these two parameters were defined empirically for each dataset/modality, as well as for each of the *ML* and *MAP* layers.

Each predicted value on the test set from the Logit layer has a score value corresponding to its bin range in the respective class histogram, which is illustrated in Fig. 4. For the *MAP* layer, the prior is modeled by a Gaussian distribution that guarantees a smoother distribution of the prediction values, as observed within the second column of Fig. 3. Thus, $P(\mathbf{c}) \sim \mathcal{N}(\mathbf{x}|\mu, \sigma^2)$ with mean μ and variance σ^2 is calculated per class, from the training set. The modeling with different distribution techniques, Gaussian distribution and normalized histogram, aims to capture complementary information from the training data, where the maximum values per classes in the normalized histograms are different from the maximum values of the Gaussian distributions (Fig. 3).

The normal distribution is feasible for modeling an unknown distribution because it has a maximum entropy. Thus, the greater entropy can guarantee a more informative distribution and at the same time less confident information around the mean, that is, it contributes to the reduction of the overconfidence inferences. Defining otherwise, the events most likely to happen have low information content *i.e.*, low entropy. Therefore, a Gaussian distribution was defined for prior $P(c_i)$ to express a high degree of uncertainty[2] in the value of variable $\mathbf{c}$ before observing the data. Furthermore, a prior distribution with high entropy is said to be a prior distribution with high variance [63].

[2] The amount of uncertainty can be quantified, for example, using Shannon's entropy for a probability distribution.

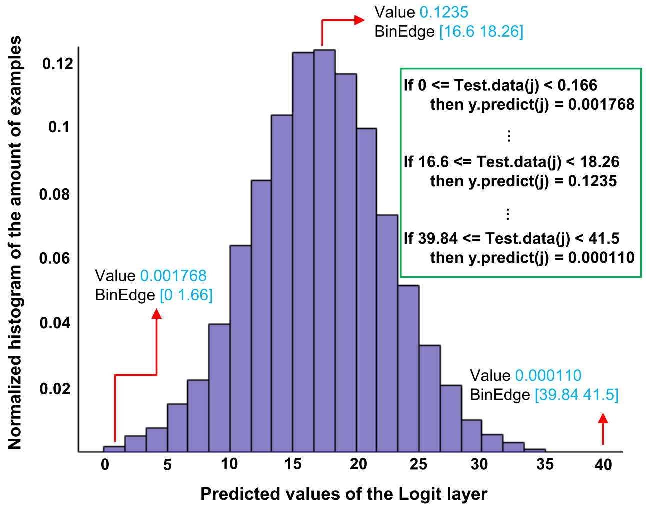


FIGURE 4. Obtaining probability values of a normalized histogram generated with the training data of the Logit layer.

Additionally, to avoid the 'zero' probability problem, as well as to incorporate some uncertainty level in the final prediction, the Additive Smoothing method (λ) [65]–[67] (also defined as Laplace smoothing) is implemented during the *ML* and *MAP* predictions. The values assigned for the Additive Smoothing are shown in Table 1, does not depend on previous information of the training set. This value was determined empirically *i.e.*, by observing which value would preserve approximately the 'original' distribution without compromising the final result. The probability estimates with the Additive Smoothing are shown in (5) and (6), *i.e.*, a small correction is incorporated into the *ML* and *MAP* estimate. Consequently, no prediction will have a 'zero' probability, no matter how unlikely.

ML layer is straightforwardly calculated by normalizing $P(\mathbf{x}|\mathbf{c})$ by the $P(\mathbf{x})$ during the prediction phase, as in (5), since the priors $P(\mathbf{c})$ are set uniformly and identically distributed for the set of classes $\mathbf{c}$,

$$ML = arg \max_i \frac{(P(\mathbf{x}|c_i) + \lambda)}{\sum_{i=1}^{nc}(P(\mathbf{x}|c_i) + \lambda)}. \tag{5}$$

Alternatively, the inference using *MAP* layer is given in (6) as follows,

$$MAP = arg \max_i \frac{(P(\mathbf{x}|c_i)P(c_i) + \lambda)}{\sum_{i=1}^{nc}(P(\mathbf{x}|c_i)P(c_i) + \lambda)}. \tag{6}$$

The sequential steps for calculating the *ML* and *MAP* is summarized within Algorithm 1, where class-conditional $P(\mathbf{x}|\mathbf{c})$ is modelled by a normalized histogram. On the other hand, to get the maximum posterior probabilities (*MAP*) the priors are modelled by normals $\mathcal{N}(test_{Lg}|\mu_{train}, \sigma^2_{train})$, where the sub-index Lg indicates that the data is obtained from the Logit layer (layer before the network prediction layer). Both the likelihood and prior are extracted from the Logit layer using the training data.[3]

```
Algorithm 1: Compute $ML$ and $MAP$
Input
  • Number of classes used in training ($nc$);
  • Number of histogram bins (nbins);
  • Values of the Logit layer on the training set ($train_{Lg}$);
  • PDF's parameters (normalized histogram and normal
    on the training set, see Fig. 3);
  • Values of the Logit layer on the testing set ($test_{Lg}$).
  • Additive smoothing ($\lambda$).
Output
  • Maximum Likelihood ($ML$) and Maximum
    a-Posteriori ($MAP$).
Getting the normalized frequency histograms:
$hc \leftarrow histogram(ScoresLogitsTrain(classes))$;
Getting the edge values of each bin of each
histogram:
$BinLow \leftarrow BinEdgesLow(hc)$;
$BinHigh \leftarrow BinEdgesHigh(hc)$;
Getting the normalized frequency values of each bin
of each histogram:
$Values \leftarrow Values(hc)$;
Getting the likelihood:
$P(\mathbf{x}|\mathbf{C}) \leftarrow zeros(size(test_{Lg}), nc)$;
$Y \leftarrow ScoresLogitsTest$;
for $k \leftarrow 1 : size(test_{Lg})$ do
    for $cla \leftarrow 1 : nc$ do
        for $i \leftarrow 1 : size(Values)$ do
            if $(BinLow(cla, i) \leqslant$
            $Y(k, cla))\,\&\,(Y(k, cla) < BinHigh(cla, i))$
            then
                $P(\mathbf{x}|C)(k, cla) \leftarrow Values(cla, i)$;
            end
        end
    end
end
Getting the Prior:
$P(\mathbf{C}) \leftarrow \mathcal{N}(test_{Lg}|[\mu_{train}, \sigma^2_{train}])$;
Calculating the $ML$ and $MAP$:
$ML \leftarrow P(\mathbf{x}|\mathbf{C}) + \lambda$;
$ML \leftarrow (ML/\mathrm{sum}(ML))$;
$MAP \leftarrow P(\mathbf{x}|\mathbf{C})P(\mathbf{C}) + \lambda$;
$MAP \leftarrow (MAP/\mathrm{sum}(MAP))$;
```

C. CNN ARCHITECTURES FOR OBJECT RECOGNITION

Experiments in [23] suggested that the greater the number of layers and neurons, the more overconfidence the result will be. However, the experiments that we have conducted show that even when reducing the amount of neurons and filters in the dense and convolutional layers, the network can still produce overconfidence in the predicted values,

[3] The code for training the network, obtaining the logit layers and computing the *ML* and *MAP* layers are available at github.com/gledsonmelotti/ML-MAP-Layers-for-Probabilistic.

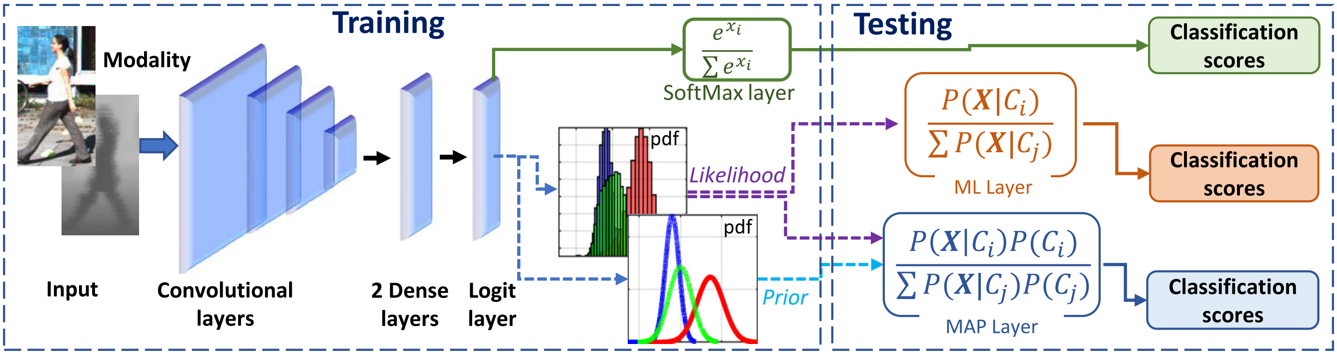


FIGURE 5. Inception V3 CNN representation with Logit and Softmax layers, Maximum Likelihood (*ML*) and Maximum a-Posteriori (*MAP*) layers. CNN's training was done with the Softmax layer. After training, the Softmax layer was replaced by the *ML* and *MAP i.e.*, the CNN was not trained with the *ML* and *MAP* layers.

as can be observed in Fig. 1. This conclusion was reached by training the Inception V3 CNN [68] and reducing the number of filters and neurons/units. Regarding object detection, the model Yolo V4 [35] was trained to detect cars, cyclists, and pedestrians, with predictions based on the SG layer.

The experiments reported throughout the remainder of this work were based on the premise that, after training the network, the proposed *ML* and *MAP* layers then replace the *SM* and *SG* prediction layers on the test set, only, according to Fig. 5.

D. RELIABILITY DIAGRAM

Typically, post-calibration predictions are analyzed in the form of reliability diagram representations [23], [69], which illustrate the relationship of the model's prediction scores in regard to the true correctness likelihood [70], as shown in Fig. 6. Reliability diagrams show the expected accuracy of the samples as a function of confidence *i.e.*, the maximum value of the prediction function.

The scores (predicted values) are grouped into M bins (histogram) in the reliability diagrams. Each sample (classification score of an object) is allocated within a bin, according to the maximum prediction value (prediction confidence). Each bin has a range $I_m = \left(\frac{(m-1)}{M}, \frac{m}{M}\right]$, where $m = 1, .., M$. The accuracy is calculated in each range I_m, as well as the average confidence $conf_{average} = \frac{1}{BM}\sum_i \hat{p}_i$, where $\hat{p}_i$ is the confidence for sample i and BM is the amount of objects in each I_m. In addition, a gap can be obtained *i.e.*, the difference between accuracy and average confidence in each range (I_m). Thus, the greater the gap, the worse the calibration result in the respective bin. Furthermore, through reliability diagrams, it is possible to obtain calibration errors, such as the Expected Calibration Error (ECE) and the Maximum Calibration Error (MCE):

$$ECE = \sum_{m=1}^{M} \frac{|BM|}{n} |acc(BM) - conf(BM)|, \tag{7}$$

$$MCE = \max_{m \in \{1,\ldots,M\}} |acc(BM) - conf(BM)|, \tag{8}$$

where n is the number of samples.

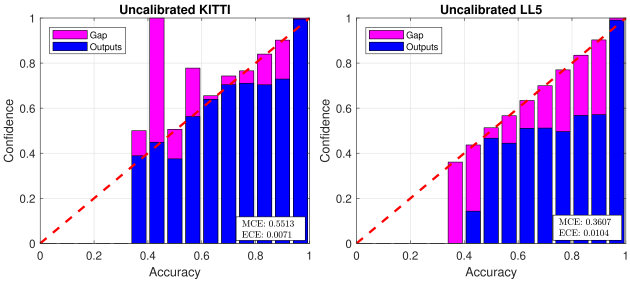


FIGURE 6. Reliability diagrams for the RGB modality on the testing set using the Softmax layer (*SM*). On the left, uncalibrated model for KITTI dataset, and on the right uncalibrated model for LL5.

TABLE 2. KITTI and LL5 dataset for classification: number of objects per class and subsets.

	KITTI dataset - 7481 Frames		
	Car	**Cyclist**	**Pedestrian**
Training	18103	1025	2827
Validation	2010	114	314
Testing	8620	488	1346
	Non-trained ('unseen-adversarial') objects		
	Tram/Truck/Van	**Tree/lamppost**	**Person-sitting**
Training	-	-	-
Validation	-	-	-
Testing	511/1094/2914	45	222
	LL5 dataset - 158757 Frames		
	Car	**Cyclist**	**Pedestrian**
Training	208501	7199	9031
Validation	23167	800	1003
Testing	193012	9238	9199
	Non-trained ('unseen-adversarial') objects		
	Bus/OtherVehicle/Truck	**Tree/lamppost**	**Motorcycle**
Training	-	-	-
Validation	-	-	-
Testing	5257/2785/10890	45	217

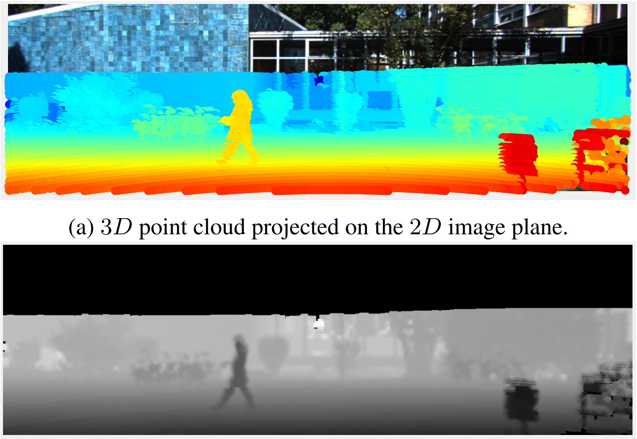
Moreover, the reliability diagrams illustrate the identity function (diagonal-dashed line) that represents a perfectly calibrated output, while any deviation from the diagonal represents a calibration error [23], [69].

E. BENCHMARKING DATASETS

A key contribution to the growing improvement of perception systems for autonomous driving is the availability of representative datasets of different modalities, such as RGB,

TABLE 3. Comparison between the classifications obtained by the *SM* layer, *ML* and *MAP* layers in terms of average F-score and *FPR* (%). The performance measures on the 'unseen' dataset are the average and the variance of the prediction scores.

			KITTI dataset			
Modalities	$\mathrm{SM_{RGB}}$	$\mathrm{ML_{RGB}}$	$\mathrm{MAP_{RGB}}$	$\mathrm{SM_{RV}}$	$\mathrm{ML_{RV}}$	$\mathrm{MAP_{RV}}$
F-score	95.89	94.85	95.07	90.29	89.70	89.50
FPR	1.60	1.21	1.19	2.73	2.64	2.60
$\mathrm{Ave.Scores_{FP}}$	0.853	0.487	0.359	0.874	0.656	0.387
$\mathrm{Var.Scores_{FP}}$	0.210	0.018	0.003	0.028	0.024	0.004
$\mathrm{Ave.Scores_{unseen}}$	0.983	0.708	0.397	0.970	0.692	0.394
$\mathrm{Var.Scores_{unseen}}$	0.005	0.025	0.004	0.010	0.017	0.003
			LL5 dataset			
Modalities	$\mathrm{SM_{RGB}}$	$\mathrm{ML_{RGB}}$	$\mathrm{MAP_{RGB}}$	$\mathrm{SM_{RV}}$	$\mathrm{ML_{RV}}$	$\mathrm{MAP_{RV}}$
F-score	92.85	92.84	92.91	90.16	89.91	89.94
FPR	2.40	2.16	1.98	2.17	1.78	1.76
$\mathrm{Ave.Scores_{FP}}$	0.939	0.531	0.383	0.914	0.574	0.398
$\mathrm{Var.Scores_{FP}}$	0.015	0.040	0.009	0.017	0.036	0.009
$\mathrm{Ave.Scores_{unseen}}$	0.996	0.454	0.375	0.996	0.502	0.374
$\mathrm{Var.Scores_{unseen}}$	0.001	0.037	0.009	0.001	0.038	0.006



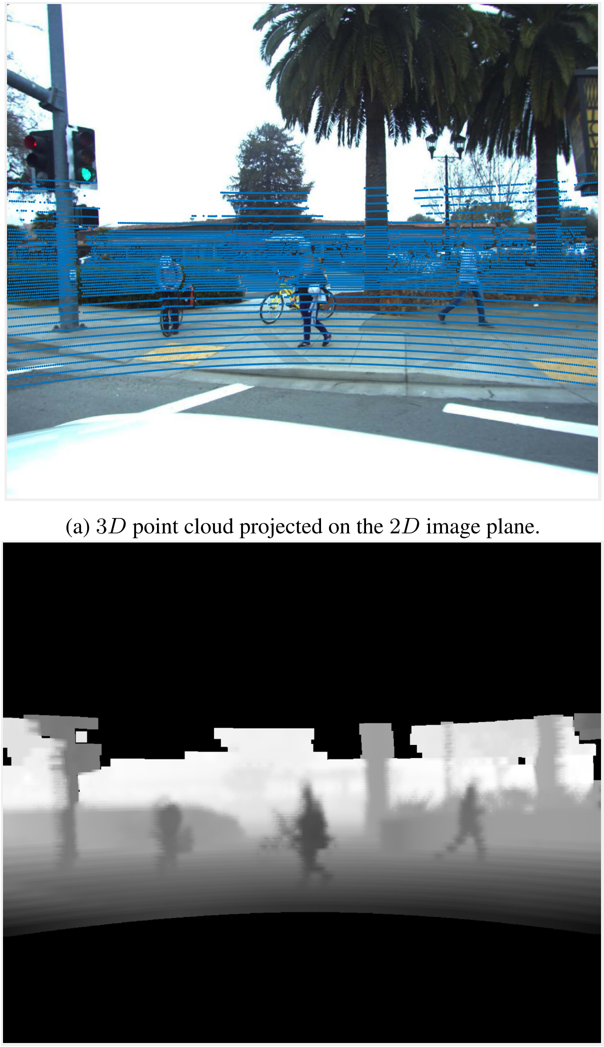
(a) $3D$ point cloud projected on the $2D$ image plane.

(b) Range-view (RV) after upsampling the point cloud.

FIGURE 7. Example from the KITTI dataset. Representations of a 'raw' point-cloud (a) in image coordinates and the upsampled range-view (b) obtained using the bilateral filter.

LiDAR, and radar [71]–[76]. In this work, we used the KITTI Vision Benchmark Suite-$2D$ object [33] and Lyft Level-5 (LL5) Perception [77], [78] datasets. The classes of interest were pedestrians, cars, and cyclists. Table 2 shows the number of objects cropped from both the RGB and range-view (depth from the LiDAR modality) images. In addition, some extra objects from the unseen/non-trained classes (not used during training), such as a person sitting, tram, truck, van, tree, lamppost, signpost, bus, and motorcycle classes were classified in the test/prediction phase, to verify the erroneous overconfidence from the prediction layers of the trained networks. Such a class can be understood as an 'adversarial' class; *Note that this research did not carry out any study involving adversarial network architectures.*

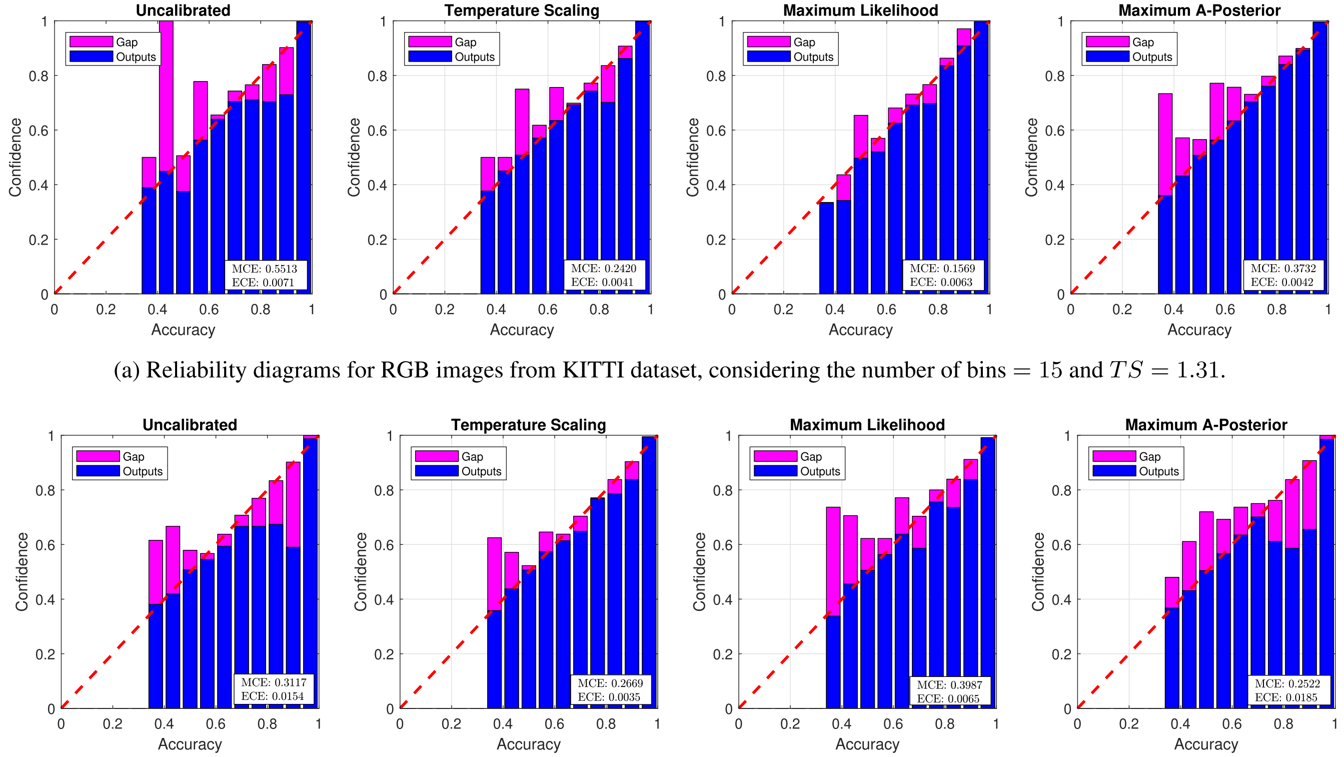
Range-view images were obtained by a coordinate transformation of the $3D$ point clouds on the $2D$ image plane followed by an upsample of the projected points. The upsample was performed using a bilateral filter, and considered a mask size 13 × 13 (sliding-windows) [34] for t he KITTI dataset and a mask size 23 × 23 for LL5 dataset. Examples of these operations can be observed in Fig. 7 and Fig. 8, respectively.



(a) $3D$ point cloud projected on the $2D$ image plane.

(b) Range-view (RV) for a 40-channels LiDAR.

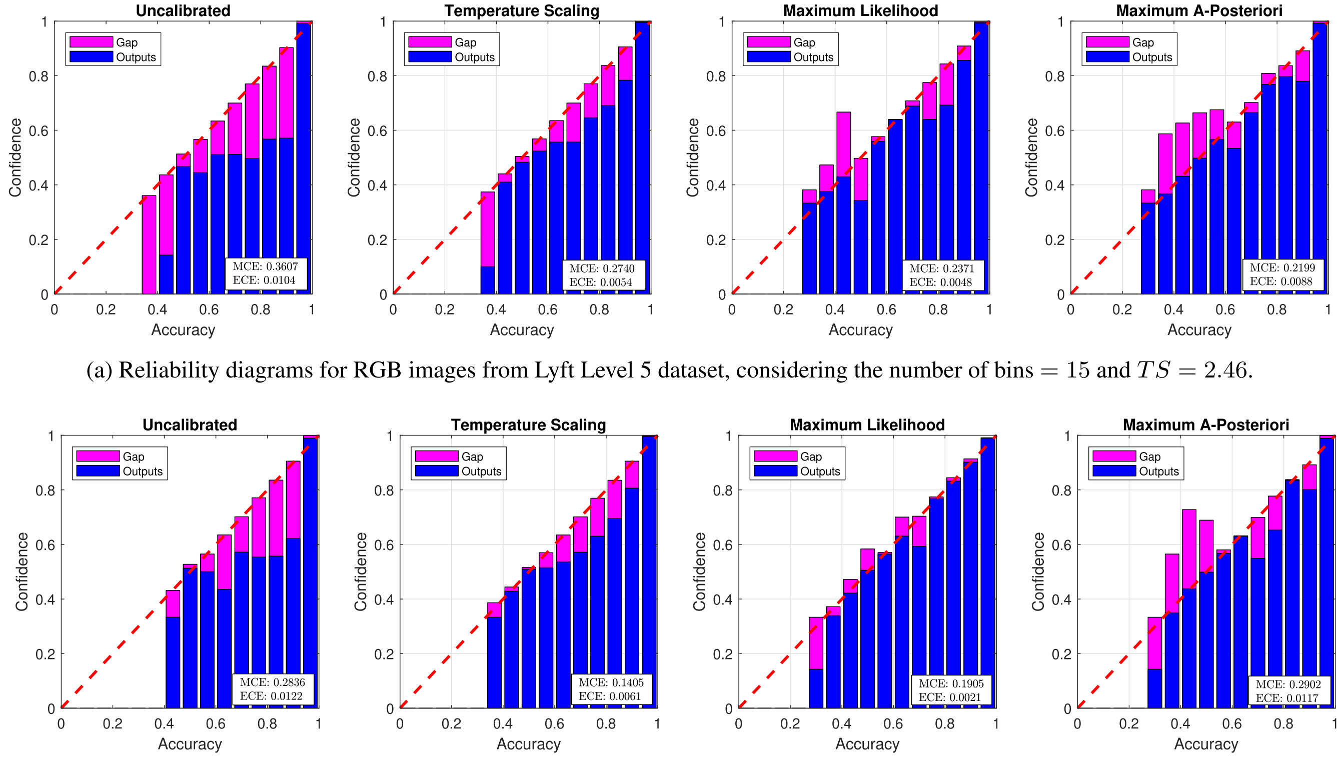
FIGURE 8. Example from the LL5 dataset. In (a) the $3D$ point clouds are in pixel-coordinates, and (b) shows the respective range-view after applying the bilateral filter.



(a) Reliability diagrams for RGB images from KITTI dataset, considering the number of bins $= 15$ and $TS = 1.31$.

(b) Reliability diagrams for RV images from KITTI dataset, considering the number of bins $= 15$ and $TS = 2.26$.

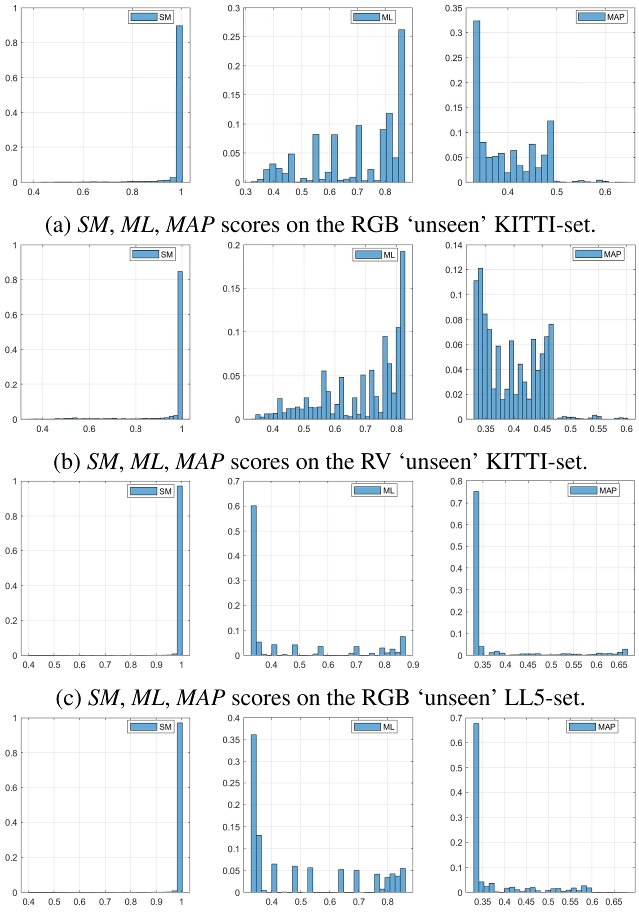
FIGURE 9. The graphs, from left to right, represent uncalibrated score values, followed by score values calibrated through Temperature Scaling, then scores obtained by the *ML* and *MAP* layers respectively.



(a) Reliability diagrams for RGB images from Lyft Level 5 dataset, considering the number of bins $= 15$ and $TS = 2.46$.

(b) Reliability diagrams for RV images from Lyft Level 5 dataset, considering the number of bins $= 15$ and $TS = 1.90$.

FIGURE 10. Reliability diagrams, on the LL5 dataset, for the following cases (from left-right): uncalibrated scores, calibrated model using TS, and then the diagrams for the models using *ML* and *MAP* layers.



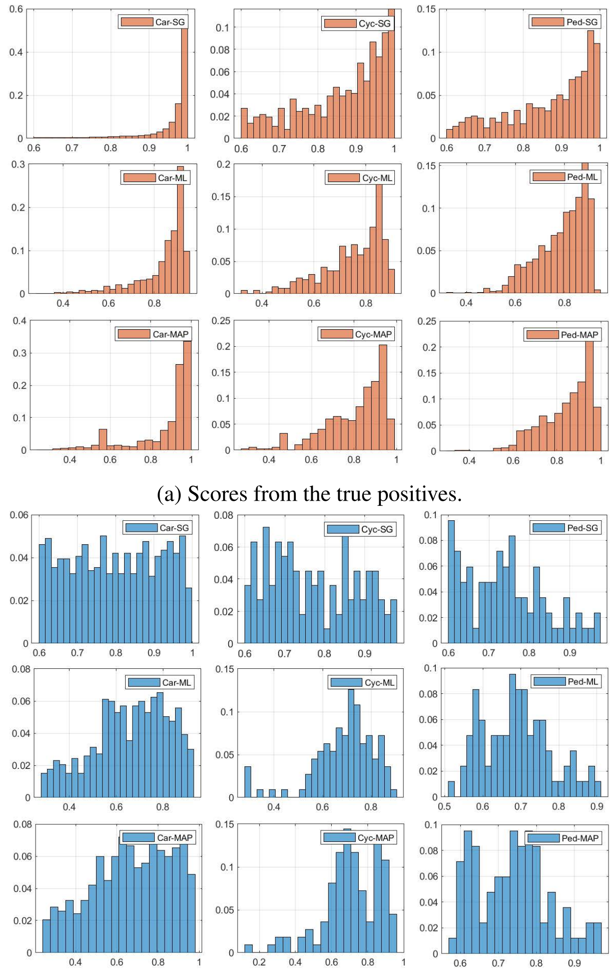
(a) *SM*, *ML*, *MAP* scores on the RGB ‘unseen’ KITTI-set.

(b) *SM*, *ML*, *MAP* scores on the RV ‘unseen’ KITTI-set.

(c) *SM*, *ML*, *MAP* scores on the RGB ‘unseen’ LL5-set.

(d) *SM*, *ML*, *MAP* scores on the RV ‘unseen’ LL5-set.

FIGURE 11. Prediction scores on the unseen/non-trained data (comprising the classes: person sitting, tram, tree/ lamppost/signpost, truck, van), using *SM* layer (left side), and the proposed *ML* (center) and *MAP* (right side) layers. The graphs of the first two rows are the results of the KITTI dataset, while the last two are from the LL5 dataset.



(a) Scores from the true positives.

(b) Scores from the false positives.

FIGURE 12. Results obtained from the Yolo V4. The columns from left to right represent the car, cyclist and pedestrian classes, as well as the distributions of the Sigmoid layer, Maximum Likelihood and Maximum a-Posteriori functions scores. The first line of the distributions are the results of the classifications of the true positives, while the last line is the corresponding scores of the false positives.

As a way to validate the proposed methodology for object detection, the KITTI Vision Benchmark Suite-2*D* object was used. The respective dataset was divided into 3367 frames for the training dataset, 375 frames for the validation dataset and 3739 frames for the test dataset.

IV. EVALUATION AND RESULTS

The output scores of the CNN indicate a degree of certainty of the given prediction. The level of certainty can be defined as the confidence of the model, and in an object recognition problem, represents the maximum value within the prediction layer. However, the output scores may not always represent a reliable indication of certainty with regard to a given class, especially when unseen (non-trained) objects occur in the prediction stage; this is particularly relevant for a real world application involving autonomous robots and vehicles, since unpredictable objects are likely to be encountered which would be misclassified by prediction layers with a high degree of certainty. With this in mind, in addition to the trained classes (pedestrian, car, and cyclist), a set of unseen objects were introduced into the classification dataset, according to Subsection III-E. Regarding the object detection, the unseen classes are already contained in the dataset's own frames. Unlike the results reported on the classification dataset, the object detection results are presented by means of precision-recall curves considering the easy, moderate, and hard cases, according to the devkit-tool provided by the KITTI benchmark.

A. RESULTS ON OBJECT CLASSIFICATION

All classes for the training dataset were extracted directly from the aforementioned datasets, except for the tree, lamppost, and signpost classes which were manually extracted from the data for this study. The rationale behind this is to evaluate the prediction confidence of the network on objects that do not belong to any of the trained classes, and as such the consistency of the models can be assessed. Ideally, if the classifiers are perfectly consistent in terms of probability interpretation, the prediction scores would be

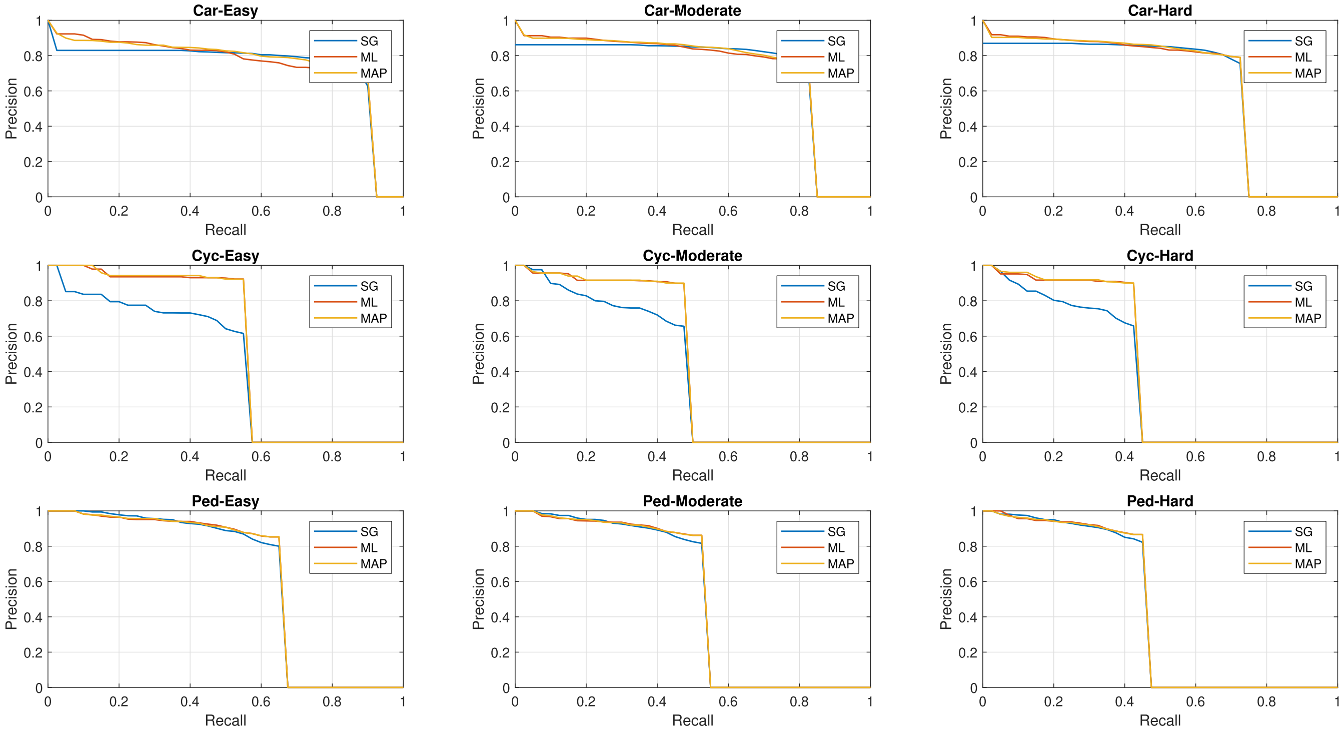


FIGURE 13. Precision-recall curves for Yolo V4 obtained from the Sigmoid prediction layer, *ML* and *MAP* layers on the KITTI dataset, considering the true positives. The curves were obtained for the easy, moderate and hard cases, according to the toolbox provided by KITTI.

identical (equal to 1/3) for each class in each sample of the unseen dataset. Results on the testing set are shown in Table 3 in terms of F-score, false positive rate (FPR), the average ($Ave.Scores_{FP}$) and variance ($Var.Scores_{FP}$) of the false positives (FP). The average ($Ave.Scores_{unseen}$) and the variance ($Var.Scores_{unseen}$) of the predicted scores are also shown for the unseen testing set (out-of-training distribution test data).

In reference to Table 3, where the results are reported based on the classification test set, it can be observed that the FPR, $Ave.Scores_{FP}$ and $Var.Scores_{FP}$ values are considerably lower than the results presented by the *SM* layer for both of the sensor modalities and datasets. Regarding the F-scores of the proposed approach (*ML* and *MAP*) compared to the *SM* resulted in an average reduction of 1% (percentage point) for the RGB modality and 0.76% for RV modality, considering KITTI dataset. The F-scores on the LL5 dataset got a gain of 0.065% for RGB modality, considering the *MAP* approach, F-score of the VR modality had a average reduction of 0.26%. Such reductions of the F-scores are relatively small and thus did not compromise the classification ability. Additionally, the distribution of the top-label scores on the test set comprising the objects that belong to the trained classes (in-distribution classes) is discussed in the Appendix VI-A.

Another way of analyzing the results of reducing overconfidence predictions is through reliability diagrams, as shown in the figures 9 and 10, considering uncalibrated, *ML* and *MAP* data. Furthermore, as a way of validating our methodology, we compared our results achieved with the temperature scaling calibration technique. Note that the results presented through the reliability diagrams are shown through the MCE and ECE metrics. From these metrics we cannot say which is the best calibration technique, because for a given technique the lowest value for the MCE was obtained, while for another technique the lowest value for the ECE was obtained. However, we show that the proposed approach contributed to reduce the calibration errors *i.e.*, to reduce the values of the MCE and ECE metrics when compared to the uncalibrated data, and consequently we provide a more reliable result, as well as the contribution to reduce the overconfidence predictions.

Further experiments have been carried out as a complementary analysis concerning the network's overconfidence behaviour, on a so-called 'unseen' test set, by means of the network's average score $Ave.Scores_{unseen}$. Note that for *ML* and *MAP* layers, the results are smaller than the *SM* layer as can be seen in Table 3. This indicates that the probabilistic inferences are significantly better balanced *i.e.*, enabling more reliable decision-making, when 'new' objects of 'non-trained' classes are presented to the CNNs, as illustrated by Fig. 11 *i.e.*, the distribution for the unseen dataset. We can see that the aforementioned graphs show less extreme results than those provided by the *SM* layer.

B. RESULTS ON OBJECT DETECTION

The results on the object detection dataset using the *ML* and *MAP* nonlinearities are impressive. Such results were not presented through reliability diagrams, but through

TABLE 4. Comparison of the areas under the curves (%) between the Sigmoid layer (SG), *ML* and *MAP* layers from the precision-recall curves.

Easy				Moderate				Hard			
	SG	**ML**	**MAP**		**SG**	**ML**	**MAP**		**SG**	**ML**	**MAP**
Car	73.62	74.29	75.18	**Car**	70.47	71.34	71.68	**Car**	62.74	63.77	63.85
Cyc	43.24	53.43	53.63	**Cyc**	39.70	45.31	45.37	**Cyc**	35.61	40.62	40.82
Ped	61.82	62.07	62.08	**Ped**	49.79	50.01	50.03	**Ped**	42.90	43.14	43.08

normalized histograms, which showed more clearly the reduction in overconfidence in relation to objects detected as false positives without degrading the results of the true positives, as showed in Fig. 12. The results are more representative through precision-recall curves, especially for the cyclist class (Cyc), whose areas under the curves (AUCs) are 24.03%, 14.28% and 14.63% for the easy, moderate and hard cases respectively, as shown in Fig. 13 and Table 4. With respect to the car (Car) and pedestrians (Ped), the proposed approach also showed some improvement.

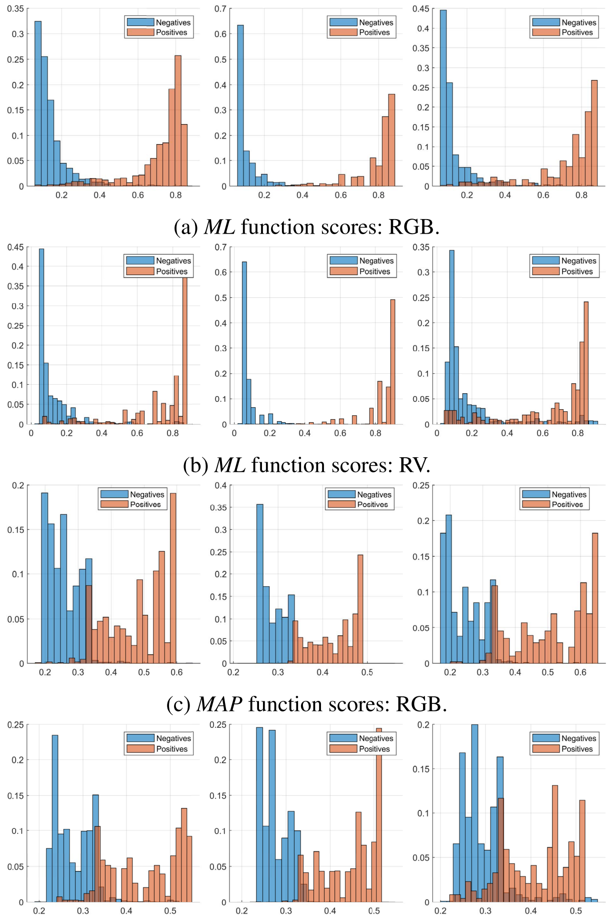
Note that the proposed methodology is dependent on the number of bins (*nbins*) and the parameter λ. Thus, the values of the scores may vary according to the values of these parameters. For the particular case of the cyclist class, the proposed methodology achieved strong classification performance compared to the baseline (results in Table 4). In this paper we have chosen to use a single set of parameters for all the three cases (*i.e.*, the same values of λ and *nbins* for each class). Given the proposed approach, we note that a set of tailored parameters for each class can be used instead, as the distributions (PDF's) are carried out individually.

V. DISCUSSION AND CONCLUSION

Within the experiments performed in this work, a probabilistic approach for CNNs was addressed as distributions in the Logit layer to better represent the classification outputs. The results reported within the experiments in this work are promising given that *ML* and *MAP* noticeably reduced the classifier overconfidence and provided a more significant distribution in terms of probabilistic interpretation.

The improvement is not as significant when analyzing objects defined as true positives. But, our concern is to develop a methodology that can reduce the values of false positives (mainly objects of the unseen class: which may be critical in robotics and autonomous driving applications) without degrading the results achieved by true positives. Note that we have included two metrics in Table 3, in order to show the reduction of score values for the 'unseen' class (in particular) and also to show that the overconfidence behavior has been mitigated for TPs and FPs.

One potential way to improve the F-scores achieved by the *ML* and *MAP* layers would be to obtain a 'perfect' match between the smoothing parameter (λ) and the number of bins in the histograms. For the new results with the EfficientNetB1 network, we have selected the parameters by using an exhaustive search process (combining several values as possible), in order to keep the values of the F-scores of the *ML* and *MAP* layers practically identical to those achieved by the EfficientNetB1 baseline. Figures 16, 17, and 18 show reductions on the scores for objects of class 'unseen' thus, the proposed approach is efficient.



(a) *ML* function scores: RGB.

(b) *ML* function scores: RV.

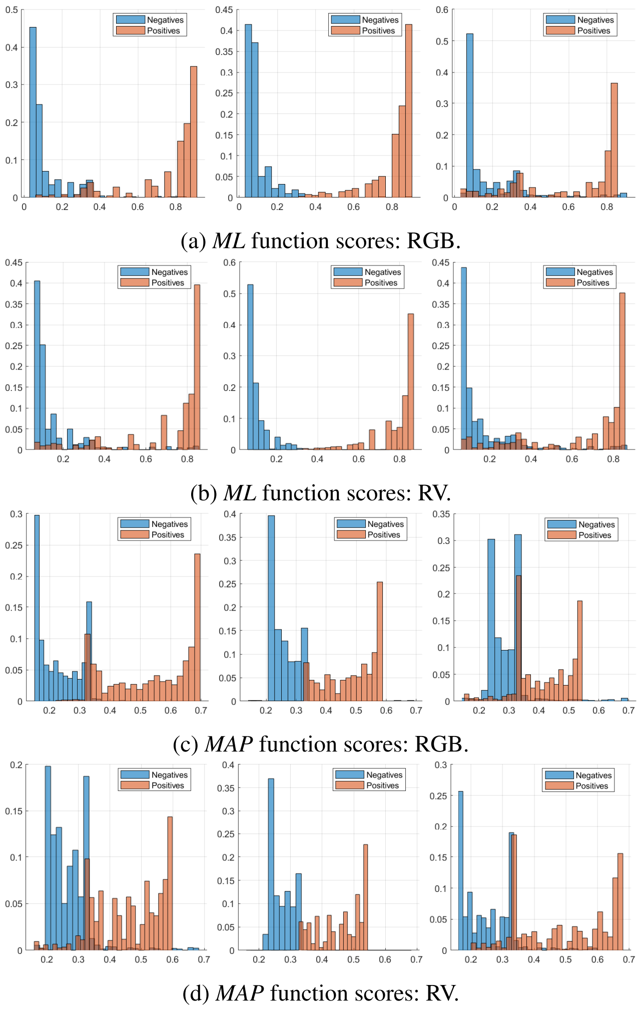
(c) *MAP* function scores: RGB.

(d) *MAP* function scores: RV.

FIGURE 14. From the RGB and LiDAR (RV) modalities, the prediction scores were calculated using the *ML* and *MAP* functions on the KITTI dataset.

As a consequence of the Additive Smoothing, the score values equal to 0.0 and 1.0 are excluded from the prediction values. The influence of the λ parameter on the data distribution can be seen from the figures in Appendix VI-B, particularly with respect to objects of the 'unseen' class.

To assess the classifier's robustness or the uncertainty of the model when predicting objects of unseen classes by the network, we considered a test set comprised of 'new' objects. Overall, the results are promising, since the distribution of the predictions were not extremities relative to the results from



(a) *ML* function scores: RGB.

(b) *ML* function scores: RV.

(c) *MAP* function scores: RGB.

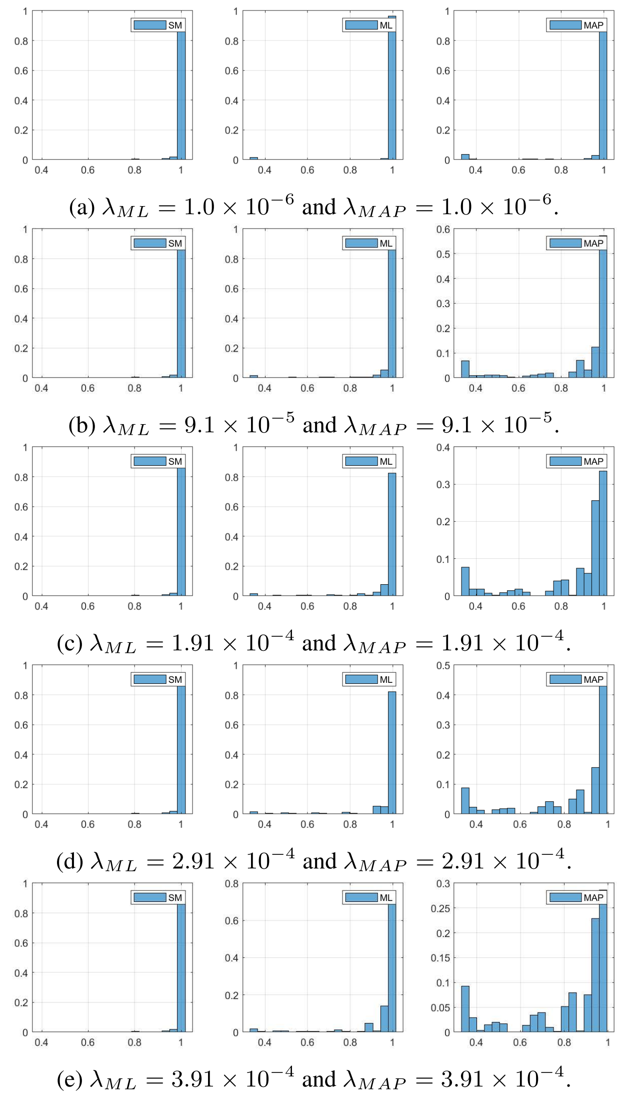
(d) *MAP* function scores: RV.

FIGURE 15. Prediction scores on the testing set for RGB and LiDAR (RV) modalities with the LL5 dataset, using the *ML* and *MAP* functions.

the *SM* layer, in other words, the average scores using *ML* and *MAP* layers were significantly lower than the Softmax prediction layer (the baseline), and thus the CNNs are less prone to overconfidence.

The results for object classification were presented through reliability diagrams, taking into account the MCE and ECE metrics. In fact, such metrics indicate how much the predicted score values are calibrated, that is, the best calibration has to present the lowest value for the MCE and ECE. However, we observed that depending on the dataset and sensor modality, our approach obtained the best result in only one of the metrics *i.e.*, either the lowest value for the MCE metric or the lowest value for the ECE metric. This fact can also be noticed with the temperature scaling calibration technique.

Another important factor that contributes to validate the proposed approach is the use of two different datasets, in terms of both RGB and Range-View (3*D* point clouds-LiDARs) modalities, since the sensors of the datasets have different resolutions, mainly the LiDAR sensor; While the KITTI dataset provides 3*D* point clouds obtained from a sensor with 64 beams, the LL5 dataset provides 3*D* point clouds with 40 beams - and so, the proposed approach was also successful with differing sensor resolutions within the state of the art.



(a) $\lambda_{ML} = 1.0 \times 10^{-6}$ and $\lambda_{MAP} = 1.0 \times 10^{-6}$.

(b) $\lambda_{ML} = 9.1 \times 10^{-5}$ and $\lambda_{MAP} = 9.1 \times 10^{-5}$.

(c) $\lambda_{ML} = 1.91 \times 10^{-4}$ and $\lambda_{MAP} = 1.91 \times 10^{-4}$.

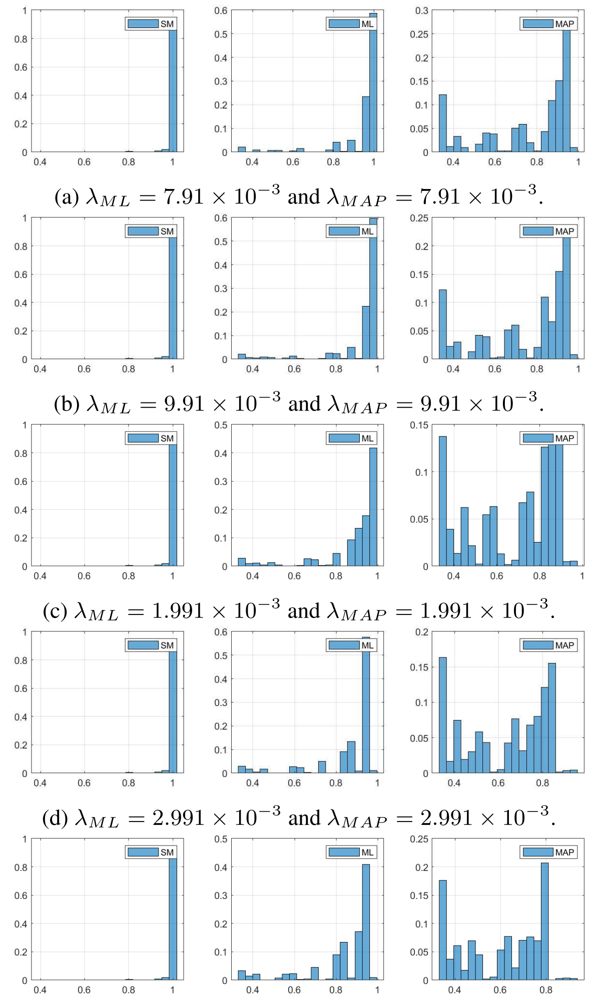
(d) $\lambda_{ML} = 2.91 \times 10^{-4}$ and $\lambda_{MAP} = 2.91 \times 10^{-4}$.

(e) $\lambda_{ML} = 3.91 \times 10^{-4}$ and $\lambda_{MAP} = 3.91 \times 10^{-4}$.

FIGURE 16. Prediction scores on the RGB unseen/non-trained data, using *SM* layer (left side), and the proposed *ML* (center) and *MAP* (right side). The *SM* case, that does not depend on λ, serves as baseline for comparison.

The proposed methodology also obtained good results for object detection, not degrading the results when compared to the *SG* prediction layer, presenting better results in all cases. The improvement is more evident for the 'cyclist' class, which contains the least amount of examples. This is an interesting result that could be further investigated in future work.

Regarding the formulations of probabilistic distributions, the prior modeling by a Gaussian distribution was shown to guarantee a smoother distribution for the prediction values. Unlike the prior, the likelihood function was modeled by means of a normalized histogram *i.e.*, by a non-parametric formulation showing the probability distributions. If both



(a) $\lambda_{ML} = 7.91 \times 10^{-3}$ and $\lambda_{MAP} = 7.91 \times 10^{-3}$.

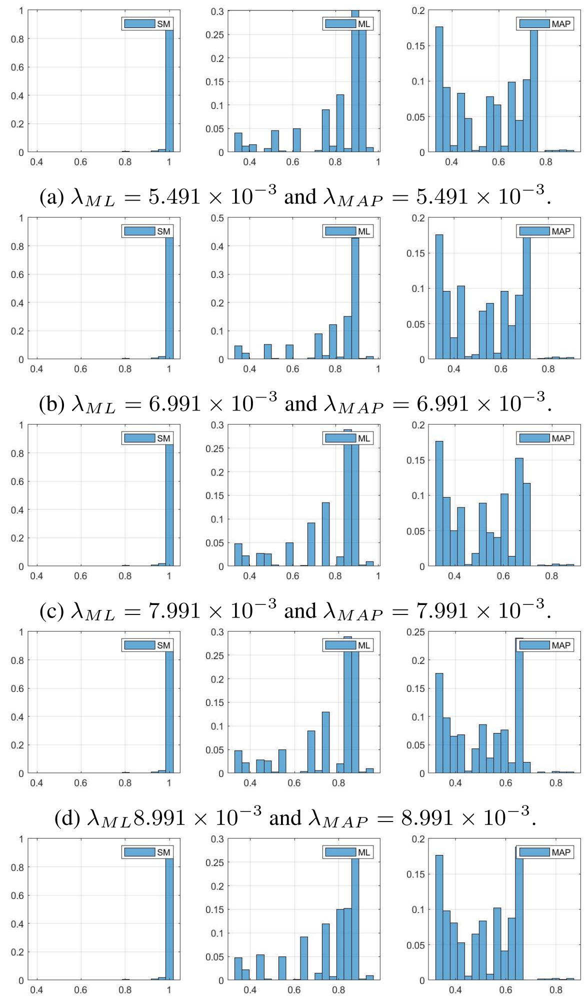
(b) $\lambda_{ML} = 9.91 \times 10^{-3}$ and $\lambda_{MAP} = 9.91 \times 10^{-3}$.

(c) $\lambda_{ML} = 1.991 \times 10^{-3}$ and $\lambda_{MAP} = 1.991 \times 10^{-3}$.

(d) $\lambda_{ML} = 2.991 \times 10^{-3}$ and $\lambda_{MAP} = 2.991 \times 10^{-3}$.

(e) $\lambda_{ML} = 3.991 \times 10^{-3}$ and $\lambda_{MAP} = 3.991 \times 10^{-3}$.

FIGURE 17. Prediction scores on the unseen data (RGB modality), for the *SM* layer (left side), and the variations in the *ML* (center) and *MAP* layers for different values of λ.



(a) $\lambda_{ML} = 5.491 \times 10^{-3}$ and $\lambda_{MAP} = 5.491 \times 10^{-3}$.

(b) $\lambda_{ML} = 6.991 \times 10^{-3}$ and $\lambda_{MAP} = 6.991 \times 10^{-3}$.

(c) $\lambda_{ML} = 7.991 \times 10^{-3}$ and $\lambda_{MAP} = 7.991 \times 10^{-3}$.

(d) $\lambda_{ML} 8.991 \times 10^{-3}$ and $\lambda_{MAP} = 8.991 \times 10^{-3}$.

(e) $\lambda_{ML} = 9.591 \times 10^{-3}$ and $\lambda_{MAP} = 9.591 \times 10^{-3}$.

FIGURE 18. Further results, in terms of the prediction scores (RGB modality), showing the influence of different values of λ on the *ML* (center) and the *MAP* (right side). The results using the *SM* layer, in the left-hand side, serves as baseline for comparison.

the prior and the likelihood function were modeled by a uniform distribution, the final result would be similar to those achieved by the *SM* and *SG* layers, since it would not offer any smoothing for the prediction values. In fact, a uniform prior or likelihood would add a constant to the training data modeling, which would have little effect on the prediction values obtained by the *ML* and *MAP*.

VI. FUTURE WORK

Softmax and Sigmoid layers represent confidence measures, but they do not provide any measure of uncertainty of the predictions. In other words, both layers mentioned previously provide a direct measure of certainty through the maximum class probability. Such layers also do not provide any information about the certainty that the model itself has about the predictions. Therefore, we address the issues of overconfident predictions and calibration techniques in this work with a focus on perception systems for autonomous vehicles. However, we realize that there is a lack of studies on how to quantify the certainty/uncertainty of predictions in relation to calibration techniques and reliability diagrams. As we verified that the MCE and ECE metrics that quantify the calibrated data through the reliability diagrams depend on the number of bins of such diagrams, that is, by changing the number of bins, the MCE and ECE metrics can provide new error values. Thus, what is the correct value of bins to ensure that a set of predictions is well calibrated?

Regardless of the methodology to reduce overconfidence predictions or capture uncertainty in predictions, how should we assess the quality of estimated uncertainty independent of calibration and regularization techniques?

Faced with such questions and based on the studies presented in the literature on computing uncertainties of predictions and of calibration and regularization techniques, we found that evaluating the quality of uncertainty estimates is still a challenge for the following reasons:

- uncertainty estimates depend on methods, which are performed by means of approximations *i.e.*, by means of inferences;
- uncertainty estimates depend on the sample size *i.e.*, the sample size can provide a certain degree of confidence that such a sample is representative;
- it is not easy to obtain a ground truth about uncertainty estimates. In fact, during our study we did not verify the ground truth about uncertainty estimates;
- study and evaluate the quality of quantitative uncertainty metrics, such as entropy, Mutual Information, Kullback-Leibler Divergence, and predictive variance.

Based on the issues mentioned above, we intend to advance research on the quality of uncertainty estimates, including the formulation of reliability diagrams, as a way to quantify the quality of uncertainty estimates.

APPENDIX

A. PREDICTION SCORES OF THE OBJECTS ON THE TESTING SET

The proposed methodology, which is based on the *ML*/*MAP* layers, aims to reduce overconfidence predictions of deep models, especially for objects classified as false positives which sometimes receive high score values of deep networks. An ideal result would be for the network to provide lower score values for the false positives *i.e.*, objects misclassified by the network, and concurrently to attain higher scores for the true positives. As a way of validating additional results on test sets, we present the Fig. 14 and Fig. 15 that contain the results for the pedestrian, car, and cyclist classes (columns from left to right), considering the scores of the objects as being positive and negative, which show smoother distributions of scores when compared to the results shown in Fig. 1.

B. SMOOTHING PARAMETER INFLUENCE

Additionally to the results presented above, we have implemented the proposed methodology on another state-of-the-art network, the EfficientNetB1. The performance achieved by the EfficientNetB1 to classify RGB images is a F-score of 98.67% using the Softmax layer (as baseline). The result achieved through the *ML* layer is equivalent to the baseline *i.e.*, F-score $=$ 98.67%, while using the *MAP* layer the network achieved 98.66% (almost the same). By keeping $nbins = 19$ for both cases, we have performed several runs by changing the values of λ, and the resulting F-score stabilized around 99.66% *i.e.*, very close to the F-score provided by the Softmax layer (baseline). A way to choose the best values for nbins and λ could be, for instance, by reducing the values of the scores of the objects classified as false positives without degrading the results of the true positives, as illustrated by figures 16, 17, and 18, where the distributions in each row were obtained through a given value for the λ parameter, considering classifications from the unseen dataset. Note that as the value of λ increases, the distributions tend to move away from the extreme values (0.0 and 1.0).

REFERENCES

[1] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, no. 1–3, pp. 1–308, 2020.

[2] S. Liu, L. Li, J. Tang, S. Wu, and J.-L. Gaudiot, "Creating autonomous vehicle systems," *Synth. Lectures Comput. Sci.*, vol. 6, no. 1, pp. i–186, Oct. 2017.

[3] T. Hehn, J. Kooij, and D. Gavrila, "Fast and compact image segmentation using instance stixels," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 1, pp. 45–56, Mar. 2022.

[4] Z. Wang, D. Feng, Y. Zhou, L. Rosenbaum, F. Timm, K. Dietmayer, M. Tomizuka, and W. Zhan, "Inferring spatial uncertainty in object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5792–5799.

[5] P. Cai, Y. Sun, H. Wang, and M. Liu, "VTGNet: A vision-based trajectory generation network for autonomous vehicles in urban environments," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 419–429, Sep. 2021.

[6] M. Schutera, M. Hussein, J. Abhau, R. Mikut, and M. Reischl, "Night-to-day: Online image-to-image translation for object detection within autonomous driving by night," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 480–489, Sep. 2021.

[7] H. Pan, Z. Wang, W. Zhan, and M. Tomizuka, "Towards better performance and more explainable uncertainty for 3D object detection of autonomous vehicles," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7.

[8] X. Cai, M. Giallorenzo, and K. Sarabandi, "Machine learning-based target classification for MMW radar in autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 4, pp. 678–689, Dec. 2021.

[9] C. Zhou, Y. Liu, Q. Sun, and P. Lasang, "Vehicle detection and disparity estimation using blended stereo images," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 4, pp. 690–698, Dec. 2021.

[10] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng, "A multimodality fusion deep neural network and safety test strategy for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 310–322, Jun. 2021.

[11] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[12] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang, "Segmenting objects in day and night: Edge-conditioned CNN for thermal image semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3069–3082, Jul. 2021.

[13] Z. Zuo, X. Yang, Z. Li, Y. Wang, Q. Han, L. Wang, and X. Luo, "MPC-based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 513–522, Sep. 2021.

[14] M. M. D. Santos, J. E. Hoffmann, H. G. Tosso, A. W. Malik, A. U. Rahman, and J. F. Justo, "Real-time adaptive object localization and tracking for autonomous vehicles," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 450–459, Sep. 2021.

[15] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 631–648.

[16] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3179–3189.

[17] V. S. Raudys, R. Somorjai, and R. Baumgartner, "Reducing the overconfidence of base classifiers when combining their decisions," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2003, pp. 65–73.

[18] K. B. Bulatov and D. V. Polevoy, "Reducing overconfidence in neural networks by dynamic variation of recognizer relevance," in *Proc. ECMS Edited Valeri M. Mladenov, Petia Georgieva, Grisha Spasov, Galidiya Petrova*, May 2015, pp. 488–491.

[19] A. Kristiadi, M. Hein, and P. Hennig, "Being Bayesian, even just a bit, fixes overconfidence in ReLU networks," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 5436–5446.

[20] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13888–13899.

[21] C. Gupta, A. Podkopaev, and A. Ramdas, "Distribution-free binary classification: Prediction sets, confidence intervals and calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 3711–3723.

[22] C. Gupta and A. K. Ramdas, "Top-label calibration and multiclass-to-binary reductions," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–37.

[23] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1321–1330.

[24] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[25] J. Mena, O. Pujol, and J. Vitrià, "A survey on uncertainty estimation in deep learning classification systems from a Bayesian perspective," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–35, Dec. 2022.

[26] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proc. 18th Int. Conf. Mach. Learn.*, San Mateo, CA, USA: Morgan Kaufmann, 2001, pp. 609–616.

[27] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Binary classifier calibration: Non-parametric approach," 2014, *arXiv:1401.3390*.

[28] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017, *arXiv:1701.06548*.

[29] K. Posch and J. Pilz, "Correlated parameters to accurately measure uncertainty in deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1037–1051, Mar. 2021.

[30] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, "Confidence regularized self-training," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5981–5990.

[31] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," 2021, *arXiv:2107.03342*.

[32] M. Martin, A. Roitberg, M. Haurilet, M. Horne, S. ReiB, M. Voit, and R. Stiefelhagen, "Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2801–2810.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[34] G. Melotti, C. Premebida, and N. Goncalves, "Multimodal deep-learning for object recognition combining camera and LIDAR data," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2020, pp. 177–182.

[35] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[36] G. Melotti, C. Premebida, J. J. Bird, D. R. Faria, and N. Gonçalves, "Probabilistic object classification using CNN ML-MAP layers," in *Proc. Workshop Perception Auto. Driving, Eur. Conf. Comput. Vis.*, 2020, pp. 1–7.

[37] G. Melotti, W. Lu, D. Zhao, A. Asvadi, N. Gonçalves, and C. Premebida, "Probabilistic approach for road-users detection," 2021, *arXiv:2112.01360*.

[38] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to Bayesian convolutional neural network with variational inference," 2019, *arXiv:1901.02731*.

[39] A. Graves, "Practical variational inference for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2348–2356.

[40] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6402–6413.

[41] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[42] R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, and A. Weller, "Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4745–4753.

[43] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging heteroscedastic aleatoric uncertainties for robust real-time LiDAR 3D object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1280–1287.

[44] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3D vehicle detection," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3266–3273.

[45] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn. (PMLR)*, vol. 48, 2016, pp. 1050–1059.

[46] X. Jia, J. Yang, R. Liu, X. Wang, S. D. Cotofana, and W. Zhao, "Efficient computation reduction in Bayesian neural networks through feature decomposition and memorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1703–1712, Apr. 2021.

[47] A. Y. Ng, "Feature selection, L_1 vs. L_2 regularization, and rotational invariance," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 78.

[48] M. Lukasik, S. Bhojanapalli, A. Menon, and S. Kumar, "Does label smoothing mitigate label noise?" in *Proc. 37th Int. Conf. Mach. Learn. (PMLR)*, vol. 119, 2020, pp. 6448–6458.

[49] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, 2015, pp. 1–9.

[50] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 2898–2909.

[51] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," 2018, *arXiv:1802.04865*.

[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.

[53] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[55] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, vol. 28, May 2013, pp. 1058–1066.

[56] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 694–699.

[57] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 2000.

[58] M. Kull, T. Silva Filho, and P. Flach, "Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers," in *Proc. 20th AISTATS*, 2017, pp. 623–631.

[59] J. Zhang, B. Kailkhura, and T. Han, "Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1117–11128.

[60] Q. Chen, W. Zhang, J. Yu, and J. Fan, "Embedding complementary deep networks for image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9230–9239.

[61] Y. Liang, H. Huang, Z. Cai, Z. Hao, and K. C. Tan, "Deep infrared pedestrian classification based on automatic image matting," *Appl. Soft Comput.*, vol. 77, pp. 484–496, Apr. 2019.

[62] A. Papoulis and U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, Nov. 2001.

[63] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.

[64] D. W. Scott, *Multivariate Density Estimation : Theory, Practice, and Visualization* (Wiley Series in Probability and Mathematical Statistics). Hoboken, NJ, USA: Wiley, 1992.

[65] D. Valcarce, J. Parapar, and Á. Barreiro, "Additive smoothing for relevance-based language modelling of recommender systems," in *Proc. 4th Spanish Conf. Inf. Retr.*, Jun. 2016, pp. 1–8.

[66] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Comput. Speech Lang.*, vol. 13, no. 4, pp. 359–394, 1999.

[67] G. J. Lidstone, "Note on the general case of the Bayes-laplace formula for inductive or *a posteriori* probabilities," *Trans. Fac. Actuaries*, vol. 8, pp. 182–192, Nov. 1920.

[68] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[69] M. Kull, M. P. Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12316–12326.

[70] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 625–632.

[71] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[72] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin, "A 3D dataset: Towards autonomous driving in challenging environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2267–2273.

[73] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "ApolloCar3D: A large 3D car instance understanding benchmark for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5447–5457.

[74] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "DrivingStereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 899–908.

[75] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9552–9557.

[76] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5000–5009.

[77] L. Vincent, P. Ondruska, A. Jain, S. Omari, and V. Shet, "Tutorial: Perception, prediction, and large scale data collection for autonomous cars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2019, p. 1.

[78] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. (2019). *LYFT Level 5 Perception Dataset 2020*. [Online]. Available: https://level5.lyft.com/dataset/

GLEDSON MELOTTI received the bachelor's degree in electrical engineering from the Federal University of Sao Joao del-Rei, Brazil, in 2006, and the master's degree in electrical engineering from the Federal University of Minas Gerais, Brazil, in 2009. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Coimbra, Portugal. His research interests include confidence calibration, deep learning, point clouds, and sensor fusion strategies applied to autonomous driving perception.

CRISTIANO PREMEBIDA worked as a Lecturer in autonomous vehicles with the AAE Department, Loughborough University, U.K., from September 2018 to December 2019. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Coimbra, Portugal, where he is a member of the Institute of Systems and Robotics (ISR-UC). His main research interests include robotic perception, machine learning, Bayesian inference, autonomous vehicles, autonomous robots, agricultural robotics, and sensor fusion. He works on multimodal and multisensory perception for robotics and autonomous systems applications, developing calibration strategies, and probability-prediction approaches to increase robustness of deep models.

JORDAN J. BIRD received the Ph.D. degree in human-robot interaction from Aston University. He is currently a Research Fellow with the Computational Intelligence and Applications Research Group (CIA), Department of Computer Science, Nottingham Trent University, U.K. His research interests include artificial intelligence (AI), human–robot interaction (HRI), machine learning (ML), deep learning, transfer learning, and data augmentation.

DIEGO R. FARIA received the bachelor's degree in informatics technology (data computing and information), in 2001, the M.Sc. degree in computer science from the Federal University of Parana, Brazil, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Coimbra, Portugal, in 2014. He has finished a computer science specialization at Londrina State University, Brazil, in 2002. From 2014 to 2016, he was a Postdoctoral Fellow at the Institute of Systems and Robotics, University of Coimbra, where he collaborated on different projects funded by EU commission and the Portuguese government in areas of robot grasping, artificial perception, cognitive robotics (HRI), assistive technology and applied machine learning, including Bayesian inference. From July 2016 to February 2022, he was a Lecturer and a Senior Lecturer (from 2019) at Aston University, U.K. He is currently a Reader (Associate Professor) in robotics and adaptive systems. He is also with the School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, U.K. He is also the Co-ordinator of the EU CHIST-ERA InDex Project (Robot In-hand Dexterous manipulation by extracting data from human manipulation of objects to improve robotic autonomy and dexterity) funded by EPSRC U.K., during 2019–2022. He is also a PI of projects with industry (KTP-Innovate U.K. Scheme) related to perception and autonomous systems applied to autonomous vehicles, during 2020–2022.

NUNO GONÇALVES (Member, IEEE) received the Ph.D. degree in computer vision from the University of Coimbra, Portugal, in 2008. Since 2008, he has been a Tenured Assistant Professor with the Department of Electrical and Computers Engineering, Faculty of Sciences and Technologies, University of Coimbra. He is currently a Senior Researcher with the Institute of Systems and Robotics, University of Coimbra. He has been recently coordinating several projects centered on the technology transfer to the industry. In 2018, he joined the Portuguese Mint and Official Printing Office (INCM), where he coordinates innovation projects in areas, such as facial recognition, graphical security, information systems, and robotics. He has been working in the design and introduction of new products as result of the innovation projects. He is the author of several papers and communications in high-impact journals and international conferences. His scientific career has been mainly developed in the fields of computer vision, visual information security, and robotics, but also in computer graphics.