Safety Monitoring of Neural Networks Using Unsupervised Feature Learning and Novelty Estimation

Arian Ranjbar, Sascha Hornauer, Jonas Fredriksson, Stella X Yu, Ching-Yao Chan

Safety Monitoring of Neural Networks Using Unsupervised Feature Learning and Novelty Estimation

Arian Ranjbar, Sascha Hornauer, Jonas Fredriksson, Stella X. Yu, Ching-Yao Chan

***Abstract*—Neural networks are currently suggested to be implemented in several different driving functions of autonomous vehicles. While showing promising results the drawback lies in the difficulty of safety verification and ensuring operation as intended. The aim of this paper is to increase safety when using neural networks, by proposing a monitoring framework based on novelty estimation of incoming driving data. The idea is to use unsupervised instance discrimination to learn a similarity measure across ego-vehicle camera images. By estimating a von Mises-Fisher distribution of *expected* ego-camera images they can be compared with *unexpected* novel images. A novelty measurement is inferred through the likelihood of test frames belonging to the expected distribution. The suggested method provides competitive results to several other novelty or anomaly detection algorithms on the CIFAR-10 and CIFAR-100 datasets. It also shows promising results on real world driving scenarios by distinguishing novel driving scenes from the training data of BDD100k. Applied on the identical training-test data split, the method is also able to predict the performance profile of a segmentation network. Finally, examples are provided on how this method can be extended to find novel segments in images.**

***Index Terms*—Autonomous Vehicles, Machine Learning, Safety Systems, Monitoring.**

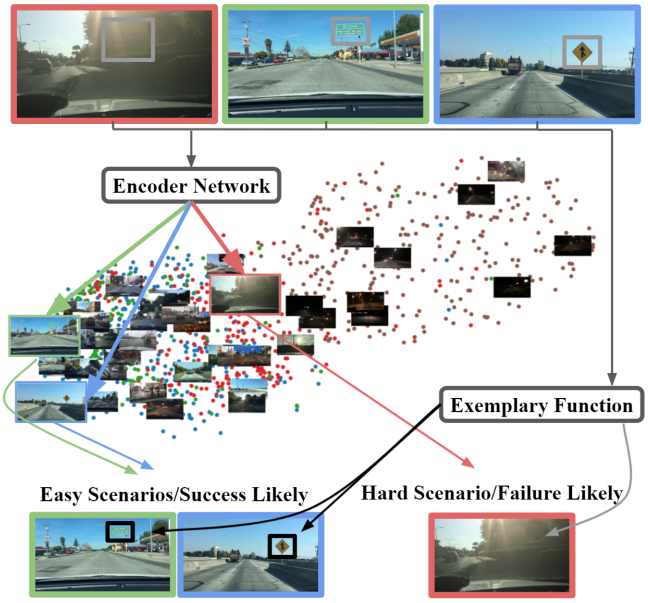

Fig. 1. The suggested method trains a CNN backbone through instance discrimination, maps all training data into the feature space of the CNN and estimate a distribution for the training data. During testing a queried image is mapped into the same feature space and the likelihood of it belonging to the training distribution is calculated. The likelihood is then used to determine the novelty of the image. The idea is to run the novelty estimation in parallel with other driving functions, as a safety measure, estimating the reliability of the predictions or decisions given by those functions.

I. Introduction

Automated vehicles are expected to drastically change the transportation industry, bringing economical, societal and in particular safety benefits. With highly automated vehicles, studies indicate that up to 50% of all road fatalities can be prevented, [1]. However, in order to introduce fully autonomous vehicles into the market they need to reach an adequate level of safety, both for the user and the surrounding traffic participants. This proves to be difficult, not only because of uncertain intentions of other traffic participants but also because of noisy sensor measurements of the surroundings leading to a potentially false ego-state perception.

The traditional approach for validating automotive safety functions is to statistically prove the validity by real world driving, covering a very high mileage, in order to get confidence in the safety argumentation, [2]. This works for supervised driving where lower safety requirements are sufficient, since the driver is responsible for safety monitoring. As the trend towards unsupervised driving is moving forward, this puts stricter requirements on the safety guarantees, implying that to use similar methods, i.e. real world driving, will require unfeasible amounts of driving data, in the magnitude of hundreds of millions of kilometers, see e.g., [2] and [3]. Researchers in the automotive safety area are developing methods to avoid this, and the proposed methods found in the literature can mainly be divided into two categories, analytical and statistical methods.

Analytical methods are used to mathematically guarantee safety constraints. These methods require a mathematical description of the system. In order to make these computationally viable, the models are made simple, in most cases meaning linear and only containing a few state variables to represent the system. As unsupervised driving systems usually are of high complexity, this limits the use of the analytical methods. To make these methods more applicable, the system can be divided into subsystems, which handle different tasks of driving. For example perception, using one or several sensors to build a model of the surroundings, see e.g. [4] or [5]; planning, building both the long term and short term trajectory for the vehicle to follow based on the model from the per-

 DOI: 10.1109/TIV.2022.3152084

ception, [6]; and control, executing the commands necessary for the vehicle to follow the plan, see e.g. [7] or [8]; where each task is analyzed individually. Other examples include division into driving modes or Operational Design Domains (ODDs), see [9]. The planning and control tasks can often be verified by defining the corresponding model in a formal language and then apply formal methods, proving that a failure state is possible or impossible, see e.g. [10], [11] or [12] for a general survey. Another technique commonly used in trajectory planning is reachability analysis, where safety is guaranteed by looking into all possible future scenarios of the ego vehicle and surrounding traffic participants, ensuring no intersections, [13].

Data driven, learning or other models of high complexity on the other hand are hard to simplify enough where traditional methods are viable. This includes deep learning methods commonly used in many of the subsystems of an automated vehicle, and in particular for perception. Although some attempts have been made to formally verify neural networks they are typically done for simple models with a few parameters, few layers or in other ways limited architectures, see e.g [14]. Instead safety arguments again rely on statistics, often requiring a lot of resources due to extensive data collection and annotation. Although data augmentation and large-scale simulation of scenarios can be done, all possible configurations of the world can never be captured. Methods such as neural networks may also give false results with high confidence even in environments or domains they were not trained in or designed for, as shown in [15].

Lately complementary approaches to the traditional verification methods have been studied using monitoring systems, running in parallel to the driving functions. For tasks such as control and planning this may involve monitoring the vehicle states in order to reject trajectories during high levels of input sensor noise resulting in unsafe motion planning, see e.g. [16] and [13]. Similarly monitoring methods have been developed for deep learning approaches rejecting predictions when considered unreliable due to the environmental conditions. This can be achieved by comparing the similarity of the current driving environment compared to the data used to train the algorithms, [17]. By utilizing an unsupervised framework for such comparison the analysis can be performed independently of the architecture of the driving function.

This paper presents a framework for novelty detection monitoring of driving environments for automated vehicles using unsupervised learning. The idea of the monitoring framework is to determine whether the vehicle has been exposed to similar driving environments before or not, as illustrated in Figure 1. If novelty is detected, predictions made by the systems are less trustable and control decisions should be made more conservative. The framework is built using Convolutional Neural Network (CNN) backbones trained unsupervised via a non parametric softmax, [18]. This paper gives a detailed description of the novelty detection framework. The main contribution is an extension of the method developed by the authors in [19] providing in-depth analysis and a better clustering approach to the original framework used to estimate the novelty of the current driving environment.

In summary the contributions are:

- A novelty estimation framework using unsupervised feature learning [19], further developed utilizing a better clustering approach and metric for the novelty estimation, increasing the performance. This also allows the method to be extended into finding novel segments of an image.
- Competitive results on pure anomaly detection on benchmarking datasets (such as CIFAR-100 [20]) while still being scalable to large driving datasets (such as BDD100k [21]) for autonomous driving.
- An investigation into the correlation between the novelty estimation and a typical driving function, e.g. segmentation using SegNet [22].
- Examples of applying the method to find novel segments of an image.

The paper is organized in the following way. Section II gives an overview of novelty estimation and the methodologies the framework is based on. It also positions the framework in relation to previous research. In section III, the framework is presented in detail and in Section IV experimental results are presented. Section V gives a brief discussion on the results, the applicability of the framework and its limitations. Finally section VI gives some concluding remarks.

II. Related Work

In this section a brief review of uncertainty estimations are given, the complementary use of monitoring systems and how they relate to novelty detection. The common use of autoencoders for novelty or anomaly detection will be explored, and also alternatives such as Generative Adversarial Networks (GANs) and metric learning. In particular why the latter is suitable for this application, including that it can be extended to process segments in addition to images.

Several attempts have been made on quantifying uncertainty in neural networks, using different types of approaches. Some focus on architectural changes as with Bayesian neural networks, [23], or ensemble networks, [24]. Others focus on modifying the the training procedure, using re-sampled training datasets, [25], or Bayesian dropout, [26]. These types of methods have all showed promising results in estimating uncertainty, however, they all still operate within the training domain. Similar techniques have also been developed to detect adversarial changes in input data. Successful attempts such as using influence functions, [27], tracing predictions back to the training data gives better results operating within the training domain than when queried with out-of-distribution examples. Temperature scaling, [28], also works on the input data by adding small perturbations, and evaluate changes of the softmax score distributions as in [29].

Rather than applying architectural changes to the acting networks, a monitoring system can run in parallel, with the task of estimating the novelty of the input data compared to the training data. This in turn can be used to estimate the reliability of the predictions given by the acting networks.

Such systems have been previously proposed to increase the safety in safety critical systems. Input reconstruction reliability estimation, [30], showed promising results reconstructing an input image from a driving network using a very limited amount of parameters. The reconstruction error could in turn be used as a reliability estimation of the network. However, due to the computational limits at the time this was only tested for low resolution images of simple test cases. This approach has later been re-investigated and extended using more modern techniques of autoencoders. In [17], an autoencoder is trained in parallel to a driving network, reconstructing the input images. Again, the reconstruction error could be used as a novelty check of the input data compared to the training data. The unsupervised nature of the autoencoder also makes it independent of what functions it monitors as opposed to [30].

Autoencoders are also widely applied within the anomaly detection domain, see e.g. [31], [32], where different architectural variants have been suggested to increase the performance. In [33], convolutional layers were used to increase the performance of anomaly detection on images. It has also been shown that density estimation in the latent space, using a Gaussian mixture model [34] or autoregressive model [35], may improve the accuracy, compared to using the reconstruction error as a measurement of novelty. In [36] a variational autoencoder [37] was implemented on driving data for novelty detection and training de-biasing for end-to-end learning, where features from the latent space were used for steering control.

Another method commonly used within the anomaly detection domain, akin to autoencoders, are GANs. Instead of an encoder and a decoder, a GAN trains a generator and discriminator network as antagonists to generate predictions with the former and judge them with the latter. In [38], the anomaly detection is based on the generators ability to represent a queried test image. By applying gradient descent on a loss function (defined by pixelwise comparison of the generated and queried image) the best feature representation in the latent space of the generator can be found. The pixelwise similarity in between the generated and queried image is then used to evaluate anomalies. Similar techniques have also been applied with different architectures of the generator and discriminator networks, see e.g. [39].

While reconstruction methods seem to have dominated the field lately, they may struggle in certain situations. For example, crucial information may be lost during deconvolution or max pooling. Because of this, more recent works have also looked into alternatives such as metric learning. In [40], a neural network is learned, mapping the training data into a feature space with the goal of enclosing the features as small as possible within a hyper-sphere. During testing, queries mapped outside of the hyper-sphere are considered anomalous.

Recently, advances within metric learning have led to significant improvements on the task of image classification, in some cases approaching supervised performance [41], [42]. In [18], a Residual Neural Network (ResNet) backbone is trained unsupervised using a non-parametric softmax layer for classification achieving state of the art results of ImageNet, [43]. Further research has also shown that similar training approaches can be extended to work for image segmentation, [44].

In this paper we will explore the use of metric learning for novelty detection. Rather than enclosing features within a hyper-sphere as in [40], unsupervised feature learning [18] will be used to map training instances onto a hyper-spherical feature space, that is the most disciriminative among them. The training data is then modeled through a von Mises-Fisher distribution and the novelty of test data is estimated through its probability of being sampled from the same distribution. The metric learning based novelty detection framework can then be used to monitor driving scenes, as have been previously suggested with the use of autoencoders, [17].

III. Novelty Estimation Framework

The idea of a novelty detection monitoring framework is to estimate the similarity of a test image y compared to the training dataset $X = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$, for some input space of images $\mathcal{X}$. In this way, the novelty of y can be determined and further indicate the expected performance of learning based methods trained on X when processing y, independent from the underlying task.

To reduce the complexity of comparing raw images an embedding function projecting onto a d-dimensional feature vector is learned $f : \mathcal{X} \mapsto \mathbb{R}^d$, which significantly reduces the dimensionality of the problem while retaining important visual information. Since no labeled examples of novel or out-of-distribution cases exist (or otherwise they would not be novel), a typical classification approach cannot be applied to train the network. Instead an unsupervised training approach is implemented using instance level discrimination, with the proxy-task of using each individual image as its own class. This results in a feature embedding that can capture visual similarities in images [18].

In section III-A and III-A1 a summary of unsupervised feature learning from [18] is given, followed by the adaption for the novelty estimation setting in section III-A2 and III-A3. An overview of the approach can be found in figure 2.

A. Unsupervised Feature Learning

Applying unsupervised feature learning, the feature embedding is constructed using a convolutional neural network f_θ, parameterized by θ. The network is trained through the proxy task of identifying each instance in the training data. In other words, each training image is assigned its own class and the task of the network is to classify each image as itself. For regular classification problems, such networks are often trained using a regular softmax layer. In this case, the probability of image x being of class c_i (corresponding to image x_i) is thus,

$$\mathbb{P}(c_i|v) = \frac{\exp(w_i^T v)}{\sum_{j=1}^{n} \exp(w_j^T v)} \tag{1}$$

where $v = f_\theta(x)$ and w_j is a weight vector for class c_j essentially serving as a prototype for the corresponding class.

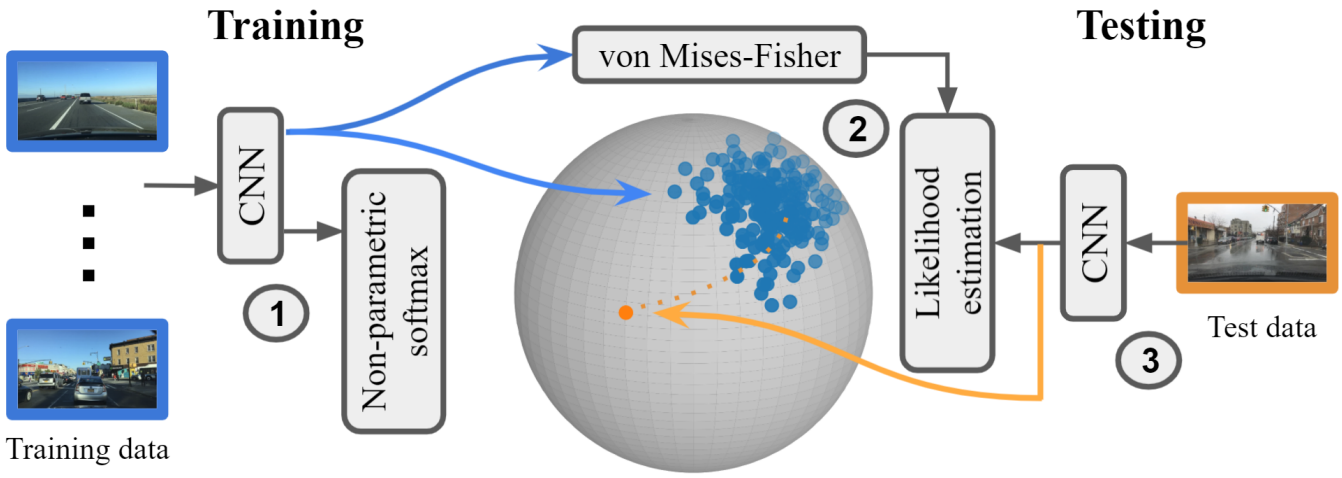

Fig. 2. In the figure the proposed framework is divided into three parts, illustrating the pipeline. First a CNN backbone is trained using non-parametric softmax with the proxy task of instance discrimination. Once the network is trained, each training image is passed through the network and the corresponding feature is extracted, before the softmax layer. Each image then has a corresponding feature in the spherical feature space, due to the normalization layer. In the second step a von Mises-Fisher distribution is estimated for the training data features. Finally in the third step testing can be done by passing a test image through the same CNN network and estimate the likelihood of the test feature being sampled from the von Mises-Fisher distribution. The likelihood is used as a measure of novelty.

In other words, $w_j^T v$ quantifies how well v matches class c_j or equivalently the similarity between x and x_j.

The fundamental problem with this setup is the similarity evaluation which is done implicitly through the prototype vectors, w_j. Each feature vector, v, is formed to be evaluated towards these prototypes, via a fully connected layer, which leaves no room for explicit comparison between features. To combat this, a non-parametric softmax [18] is implemented replacing $w_j^T v$ with $v_j^T v$ where $v_i = f_\theta(x_i)$ (using a normalization layer to ensure $\|v\| = 1$). Equation (1), can then be expressed as

$$\mathbb{P}(c_i|v) = \frac{\exp(v_i^T v/\tau)}{\sum_{j=1}^{n} \exp(v_j^T v/\tau)} \tag{2}$$

where τ controls the density of the distribution of $\{v_i\}$ on the hyper-sphere, which will show to be a vital part of the novelty estimation. Additionally, since this feature embedding allows for explicit comparison between features, it induces a metric of similarity in the corresponding feature space.

1) Training: The learning objective of the feature embedding is simply maximizing the joint probability determined from (2) over all classes (i.e. images), or equivalently minimizing the log-likelihood function

$$\hat{\theta} = \text{argmin}_\theta - \sum_{i=1}^{n} \log \mathbb{P}(c_i|f_\theta(x_i)). \tag{3}$$

Training such a model poses several computational challenges, in particular for large datasets. For each step it is required to calculate the collection of feature vectors for the non-parametric softmax layer calculations. This can be solved by storing all the features, $\mathcal{V} = \{v_1, v_2, ..., v_n\}$, of the training data in a memory bank. This also simplify the training procedure where the features from the previous iteration can be used during optimization of θ through stochastic gradient descent. In other words, while calculating the feature vectors for step $t+1$ (i.e. $\mathcal{V}_{t+1}$), $\mathcal{V}_t$ is used, where $\mathcal{V}_0$ is initialized as unit random vectors.

Although the memory bank significantly simplifies the calculation of the non-parametric softmax, it is still computationally inefficient when dealing with large training datasets. This is due to the calculation of the softmax (see Eq. (2)) where the normalization factor i.e. denominator $Z = \sum_{j=1}^{n} \exp(v_j^T v/\tau)$ may involve hundreds of thousands of terms since there is one class for each training image. This can be solved by using a negative sampling model where for the real data point x with feature v, m negative samples, $v_1, ..., v_m \in \mathcal{V} \setminus \{v\}$, are drawn from the memory bank. Using the true data point together with the negative sample during computation of Z would speed up the calculations when $m << |\mathcal{V}|$. However, to cover a large enough subset of $\mathcal{V}$ to give a sufficient approximation, m still needs to be large. Instead this is solved by using Noise-Contrastive Estimation (NCE), see e.g. [18] and [45],

$$J(\theta) = -(\mathbb{E}_{\mathbb{P}_d}[\log h(c_i, v; \theta)] + m\mathbb{E}_{\mathbb{P}_n}[\log(1 - h(c_i, v'; \theta))]) \tag{4}$$

where $\mathbb{P}_n$ denotes the uniform noise distribution (of negative samples) considered to be m times more common than the true samples, $\mathbb{P}$ the true data distribution, and

$$h(c_i, v) = \mathbb{P}(D = 1|c_i, v) = \frac{(c_i|v)}{P(i|v) + m\mathbb{P}_n(i)}. \tag{5}$$

The learning task now changes to discriminate between true and noise samples allowing the use of smaller values of m.

2) Improving training: When using driving data to train the feature embedding, it is possible to improve the result by extending the proxy task to incorporate actions. Since most driving data is collected through driving, the current action

taken by the driver is automatically given while collecting other sensor data. By redefining the driver scenes to not only include an image but rather a collection of images from the past and the future combined with actions, a better understanding of the novelty of the environment can be inferred, [46].

For time step t the driving scenario s_t is thus defined as,

$$s_t = \{(x_\tau, a_\tau) : \tau \in \{t - t_{\text{min}}, ..., t, ..., t + t_{\text{max}}\}\} \quad (6)$$

where $\{x_\tau\}_{\tau \in \{t-t_{\text{min}},\dots,t+t_{\text{max}}\}}$ are the $t_{\text{min}} + t_{\text{max}} + 1$ consecutive frames and $\{a_\tau\}_{\tau \in \{t-t_{\text{min}},\dots,t+t_{\text{max}}\}}$ are the corresponding actions $a_t \in \mathcal{A}$ for some action space $\mathcal{A}$. The actions are represented by action probability vectors [47], extended to the dimensionality of the output from the convolutional layer where the actions are concatenated with the image data.

In addition, when training on the present and previous data, i.e. $s_t = \{(x_\tau, a_\tau) : \tau \in \{t - t_{\text{min}}, ..., t\}\}$ the actions can be used for performance evaluation, in between training epochs, through prediction of $\{a_\tau\}_{\tau \in \{t+1,\dots,t+t_{\text{max}}\}}$. While still using instance discrimination for the actual training iterations, weighted k-nearest neighbor can be used to predict future actions on a smaller validation set, X_{val}, after each training epoch. For a validation scenario $s_j^{\text{val}} \in X_{\text{val}}$, it is first mapped into the feature space $v_j^{\text{val}} = f_\theta(s_j^{\text{val}})$. In the feature space it is compared to the current stored features from last training epoch, $\mathcal{V}$ finding the k-nearest neighbours with respect to cosine similarity as metric, $d = \cos(v_j^{\text{val}}, v_i)$ $v_i \in \mathcal{V}$. The top k nearest neighbors $\{v_i\}_{i \in \mathcal{N}_k}$ are then used to predict the future actions of s_j^{val} through weighted voting, where for action $a \in \mathcal{A}$ the weight is calculated through $w_{a^t} = \sum_{i \in \mathcal{N}_k} \exp(d/\tau) \mathbb{1}(a_i^t = a)$.

The aim of introducing this classification problem of predicting the upcoming actions of a particular test scenario, by looking into similar neighbors, is to increase the accuracy while also preventing overfitting. After the initial epochs of training the action prediction classification accuracy is used to determine model performance.

3) Novelty Estimation: The final step is to estimate the novelty of a new data point y compared to the training dataset $\{x_i\}$. Using the final feature embedding, the training data can be mapped as features $\mathcal{V} = \{v_i : v_i = f_\theta(x_i)\}$ onto the $d-1$-dimensional unit hyper-sphere, where $\dim v_i = d$. Assuming the training data is normally distributed on this sphere by the construction of the feature embedding, a von Mises-Fisher distribution can be fitted to the collection of feature vectors $\mathcal{V}$,

$$\begin{cases} \phi(v; \mu, \kappa) = C_d(\kappa) \exp(\kappa \mu^T v) \\ C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)} \end{cases} \quad (7)$$

where $\kappa \geq 0$ is the concentration parameter of the distribution (similar to how τ controls the density in the construction of the feature embedding), $\|\mu\| = 1$ is the mean directional feature vector and I_α is the modified Bessel function of the first kind where $\alpha = d/2 - 1$. Contrary to the training the distribution of training data is now considered to be known. The novelty of a new datapoint y is then estimated from the likelihood of $v = f_\theta(y)$ belonging to the training distribution, $\phi(v; \mu, \kappa) \mid_{\mu, \kappa}$.

IV. Experiments

In this section three experiments are demonstrated to benchmark the proposed method. The first experiment benchmarks the method against some state of the art anomaly detection algorithms on commonly used datasets for such tasks. The second experiment shows the potential on driving datasets using real world driving data. Finally, the third experiment compares the novelty estimation to performance degradation in a driving function. The aim of these experiments are to show that the presented method gives competitive results in benchmarks, that it can be applied to driving datasets and potential use cases in automated driving.

A. Anomaly Detection Benchmark

Using the experimental setup from [40] on CIFAR-10, [20], expanded upon in [19] for CIFAR-100 [20], one class is considered to be in distribution and the rest anomalies. In total, each dataset $\mathcal{D}$ is split into $N_\mathcal{D}$ classes, and corresponding $N_\mathcal{D}$ experiments are performed. For each class, a subset is used for training considered to belong to the true distribution of data. The rest is the used for testing to evaluate the performance of the model. By using the true labels (i.e. whether the test image belongs to the true data distribution or anomalies), the area under the receiver operating characteristic curve (AUROC) is used as performance metric. For CIFAR-100 the 20 superclasses are used as labels, instead of the 100 regular classes, to reduce the number of experiments. The benchmarking is done against a mix of conventional methods, autoencoders, GANs and metric learning based approaches, in addition to the original approach proposed in [19]. The benchmarked methods used are:

- One-Class Support Vector Machine (OC-SVM), [48]
- Deep Convolutional AutoEncoder (DCAE), [31]
- Deep AutoEncoding Gaussian Mixture Model (DAGMM), [34]
- Latent Space Autoregression (LSA), [35]
- Anomaly Detection with Generative Adversarial Network guiding marker discovery (AnoGAN), [38]
- Anomaly Detection with Generative Adversarial Networks (AD-GAN), [49]
- Unsupervised learning of the Set of Local Maxima (LM), [50]
- One-Class Deep Support Vector Data Description (SVDD), [40]
- Unsupervised Feature Learning with the Gaussian Estimation and Mahalanobis Metric (GM), [19]

1) Hyperparameters: The proposed model use a ResNet-18 [51] as backbone network, encoding a 128-dimensional feature vector for each image. The network is trained using the non-parametric softmax layer using stochastic gradient descent with momentum 0.9, batch size 32 and weight decay 4×10^{-5}. The learning rate is initialized to 0.01 and decreased by 10% every 30 epochs. The von Mises Fisher distribution estimation is done through expectation–maximization and asymptotic approximation of κ, as described in [52].

2) Results: Results on CIFAR-10 are given in Table I. Each row corresponds to one experiment where the denoted class is considered to be in distribution, and each column contains the corresponding AUROC value for the method as a classifier between in distribution and novel images in the test set. The suggested approach (abbreviated vMF-LE) outperforms the other methods in all but three classes. In a similar way, AUROC is reported for CIFAR-100 on the 20 superclasses in Table II, where the proposed method again outperforms the other methods on most classes.

B. Novelty Estimation in Driving Data

The second experiment investigates novelties in driving scenes from the Berkeley DeepDrive dataset (BDD100k), [21], showing the applicability for autonomous driving. BDD100k consists of 100k 1280x720 pixel images collected from driving in the United States. The data is mostly collected in San Francisco and New York to capture a wide variety of scenarios and attributes. Among the labels time of day, weather conditions and locations such as highway, city, parking lot etc is annotated, which makes the dataset suitable for this kind of benchmark. It also contains object labels for traffic participants and other relevant traffic objects such as traffic signs. The dataset is by default split into 70,000 images for training, 20,000 for testing and 10,000 for validation by default.

BDD100k also contains a subset BDD10k of 10k images where segmentation masks are provided. Each pixel is labeled with one of 20 classes ranging from road, vegetation and other background categories such as buildings, to traffic participants such as cars, pedestrians and motorcycles, among other things.

1) Experiment setup: As previously discussed, there are no predefined novelty scenarios for autonomous driving. In order to benchmark the method such scenarios are created by omitting certain properties from the training dataset. By querying the annotated labels for weather data and the time of day, BDD100k is divided into several subsets found in Table III. The training is done on good conditions, e.g. clear weather during daytime, and then the testing is done on both good and bad conditions, see Figure 3 for examples. The same hyperparameters are used as in the setup from the preceding experiment.

The method is benchmarked in this way against an autoencoder, since autoencoders have been the predominant technique suggested for novelty estimation within the autonomous driving domain. Using the best performing architecture of the autoencoder the comparison is made against the DCAE, [31]. The original structure is applied with 128, 64, 32 x (5x5x3)-filters in the encoder and symmetrical decoder replacing max-pooling with upsampling. The training is done in 250 epochs using mean squared error loss, batch size 32 and fixed weight decay of 10^{-6}. The reconstruction error is used as a measurement of similarity to the training data compared against the likelihood from the proposed model.

In addition, the BDD10k subset is used to show that the method works even under small semantic changes to the images. By querying the segmentation masks the dataset is divided into a training set without any pedestrians, and then a test set with and without pedestrians. The segmentation mask is used rather than the object labels to ensure that all images containing pedestrians are removed from the training set, since some training images might contain pedestrians not annotated in the object data. Finally, the action data inclusion is tested using $\mathcal{A} = \{\text{go straight, stop or slow, turn left, turn right}\}$ and three past and future frames, $t_{\text{min}} = t_{\text{max}} = 3$. The three past frames are used in the training and the three future frames are used to monitor the network performance of predicting actions.

Fig. 3. Examples of the different conditions, upper left shows clear weather during daytime, upper right rain during daytime, lower left clear weather during dusk/dawn and finally lower right shows clear weather during the night.

2) Results: The results can be found in Table IV, where AUROC is used again to report the performance of the model as a classifier between images considered to be in distribution or novel. Naturally it would be challenging to achieve high performance since there are a lot of similarities in between the classes of images, in particular rainy scenarios or scenes from dusk or dawn in relation to the clear daytime images (see Figure 3 for reference). This can be seen in particular for the night time images where higher performance is accomplished.

In the second part of the experiment, using BDD10k to exclude images containing pedestrians, an AUROC of 72.6% is achieved when images in the test dataset containing pedestrians are considered novelties. This is further increased to 79.2% when including actions in the images, as can be expected since driving scenarios where pedestrians are present most likely involve different driving behaviors.

C. Driving Task Performance Prediction

Up until now each experiment included labels in the testing phase in order to benchmark the performance. In the third experiment the novelty estimation will instead be used as an accuracy prediction and confidence estimation for a typical driving function. Using a similar experimental setup as in [19], the correlation of the novelty estimation is evaluated against a basic segmentation network, SegNet [22].

1) Experimental Setup: Similar to the previous setup the data is divided into different categories based on the labeling, this time for day and night time. As segmentation labels are

TABLE I
AVERAGE AUROC IN % (OVER 10 RUNS) FROM NOVELTY DETECTION ON CIFAR-10. THE IN-DISTRIBUTION CLASS IS NOTED IN THE LEFT COLUMN.

	OC-SVM	DCAE	ANO-GAN	DAGMM	AD-GAN	DEEP SVDD	Local Maxima	LSA	GM	vMF-LE
airplane	61.6	59.1	67.1	41.4	64.9	61.7	74.0	73.5	76.6	**80.2**
automobile	63.8	57.4	54.7	57.1	39.0	65.9	74.7	58.0	69.6	**78.3**
bird	50.0	48.9	52.9	53.8	65.2	50.8	62.8	69.0	**79.0**	71.1
cat	55.9	58.4	54.5	51.2	48.1	59.1	57.2	54.2	**74.5**	72.1
deer	66.0	54.0	65.1	52.2	73.5	60.9	67.8	**76.1**	71.9	69.5
dog	62.4	62.2	60.3	49.3	47.6	65.7	60.2	54.6	72.0	**79.8**
frog	74.7	51.2	58.5	64.9	62.3	67.7	75.3	75.1	77.9	**82.4**
horse	62.6	58.6	62.5	55.3	48.7	67.3	68.5	53.5	70.3	**77.2**
ship	74.9	76.8	75.8	51.9	66.0	75.9	78.1	71.7	77.4	**83.6**
truck	75.9	67.3	66.5	54.2	37.8	73.1	79.5	54.8	76.9	**80.2**
Average	64.8	59.4	61.8	53.1	55.3	64.8	69.8	64.1	74.6	**77.4**

TABLE II
AUROC (%) OVER THE 20 SUPERCLASSES OF CIFAR-100.

Superclass	OC-SVM	DAGMM	AD-GAN	vMF-LE
Aquatic mammals	68.0	43.4	63.1	**77.1**
Fish	63.1	49.5	54.9	**75.9**
Flowers	50.4	66.1	41.3	**73.7**
Food container	62.7	52.6	50.0	**70.3**
Fruit	59.7	56.9	40.6	**72.1**
Devices	53.5	52.4	42.8	**74.6**
Furniture	55.9	55.0	51.1	**73.8**
Insects	64.4	52.8	55.4	**77.8**
Carnivores	66.7	53.2	59.2	**74.8**
Man-made things	70.1	42.5	62.7	**81.1**
Natural scenes	**83.0**	52.7	79.8	73.0
Herbivore	59.7	46.4	53.7	**71.0**
Mammals	68.7	42.7	58.9	**74.0**
Invertebrates	65.0	45.4	57.4	**79.5**
People	50.7	57.2	39.4	**68.0**
Reptiles	63.5	48.8	55.6	**64.6**
Small mammals	68.3	54.4	63.3	**74.9**
Trees	71.7	36.4	66.7	**81.4**
Vehicles 1	50.2	52.4	44.3	**68.6**
Vehicles 2	57.5	50.3	53.0	**77.0**

TABLE III
THE SUBSETS USED IN THE EXPERIMENTS. THE NUMBER OF FRAMES AND DESCRIPTION OF EACH SUBSET ARE INCLUDED. TRAINING IMAGES WERE EXTRACTED FROM THE TRAINING PART OF BDD100K AND BDD10K, AND TESTING IMAGES FROM THE VALIDATION PART.

BDD subset	# frames	Description
Clear Daytime	training: 12,454 test: 1,764	weather: clear time: daytime
Rain	396	weather: rainy time: daytime
Dusk	778	weather: all time: dusk/dawn
Night	3,929	weather: all time: night
No pedestrian	training: 4571 test: 621	No segmented pedestrian
Pedestrian	379	Contain segmented pedestrian

TABLE IV
REPORTED AUROC IN SIX EXPERIMENTS. THE FIRST FOUR EXPERIMENTS USE A MODEL TRAINED ON CLEAR WEATHER DURING DAYTIME AND THE INTRODUCED NOVELTIES INCLUDE RAIN, DUSK/DAWN, NIGHT AND ONE ALL THREE CONDITIONS WHERE INCLUDED. THE FIFTH EXPERIMENT USE A MODEL TRAINED ON DATA WITHOUT PEDESTRIANS, WHERE IN THE TEST PEDESTRIANS ARE INTRODUCED AS NOVELTIES. FINALLY THE SAME EXPERIMENT IS DONE BUT THREE CONSECUTIVE FRAMES ARE USED INSTEAD OF A SINGLE IMAGE TOGETHER WITH ACTIONS.

Model	Rain	Dusk	Night	All	Ped.	Ped.Act.
DCAE	61.2	64.0	78.0	75.0	-	-
vMF	**75.7**	**69.0**	**80.4**	**78.3**	**72.6**	**79.2**

required, the subset of BDD100k providing this information (BDD10k) is used. 2972 of these images contain both a segmentation mask and descriptive labels such as time and weather. 829 of the images are taken during daytime and clear weather, of which 729 are used for training and 100 for testing. The same hyperparameters are used as in the previous example.

In parallell a SegNet model is trained using the architecture from [22], i.e. 13 convolutional layers in the encoder. The model is trained using stochastic gradient descent with batch size 8, learning rate 0.1 and momentum 0.9.

2) Results: In Figure 4 the novelty of the data is plotted against the segmentation loss from the SegNet model. Overall the segmentation loss is increasing the more visual dissimilar the test image is from the training data. Although there is some variance due to the architectural differences between the models, in other words the SegNet model may generalize better in some cases.

V. DISCUSSION

In this section a deeper look into the results will be given explaining some of the performance variances in between the experiments. Then a brief discussion on the limitations of the study is presented and some potential future research directions.

A. Performance

As seen in the first experiment the model performs well on CIFAR-10 and CIFAR-100. It also maintains the performance on the driving dataset BDD100k, both distinguishing between

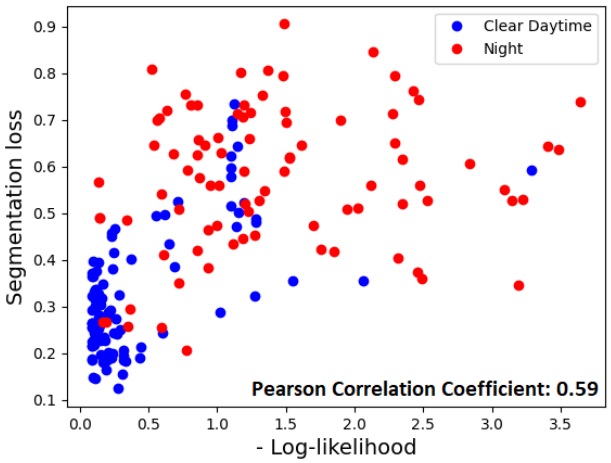

Fig. 4. Segmentation loss plotted against the negative log-likelihood of the image to belong to the true distribution. Images with higher probability of being novel generally have a higher segmentation loss.

Fig. 5. Images with ambiguous labels which impact the quantified performance. Both images in the first row are visually close to clear daytime images but are labelled as "raining, daytime" and "clear, dusk/dawn" respectively. The opposite case is shown in the second row where images are labeled as "clear, daytime" but are visually close to raining (left) and dusk/dawn (right) conditions. The third row shows examples of obvious labeling mistakes, the left being labeled as raining and the right one as night time.

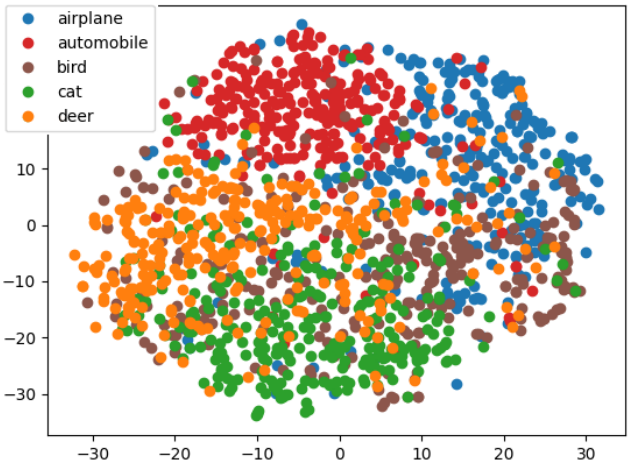

Fig. 6. T-SNE plot of training features from CIFAR-10 when using five classes for training, each represented by a different color.

different contexts in the form of weather but also smaller semantic differences such as whether pedestrians are included or not. In particular it keeps a higher relative performance compared to the autoencoder, even under less contextual changes, as in between clear weather and rain. The main difference between the two experiments is that there exists a clear semantic threshold in between the classes for CIFAR. For example, there is no doubt whether an image in CIFAR-10 is a cat or an airplane. However, for the real world driving scenarios there are several overlaps in between the classes. Consequently an experiment of this nature will never achieve as high performance as when using discrete semantic classes. To illustrate this, a few examples are gathered in Figure 5 showing the overlap between labels as well as misslabeling.

The larger spread in training data for the scenario experiment is also evident when compared to the experiment including and excluding pedestrians. In the first case, several examples of what is considered novel scenarios in the test data, are present in the training data. However, in the second case, a better split is made through the use of the segmentation mask with less overlap in between labeling, i.e. there might be a better consensus of what represents a pedestrian in contrast to clear and cloudy weather. Thereby a relatively high performance is achieved in the pedestrian/no pedestrian scenarios, even though they pose a greater challenge in theory due to less visual dissimilarity in the test set.

On the same note, pure benchmarking performance can be increased by using a multimodal mixture of von Mises-Fisher distributions, but at the cost of generality. This is evident in scenarios where underlying substructures are known, such as for classification data. In Figure 6 a t-SNE plot can be found illustrating the training features when using five classes, as being in distribution, from CIFAR-10.

Another potential performance increasing modification could be joint optimization of the feature extracting network and von Mises-Fisher distribution fitting, as in [34], but is also left out of this study.

B. Driving Scene Novelties

The experiments in section IV rely on artificially generated novelties in the test data. Instead of leaving out parts of the data, the full BDD100k dataset can be used for training. In this way examples of typical novelties to all driving data can be found. In figure 8 some of the images with least probability of belonging to the same distribution as the training data are listed, with corresponding human interpretation of why they may be considered novelties.

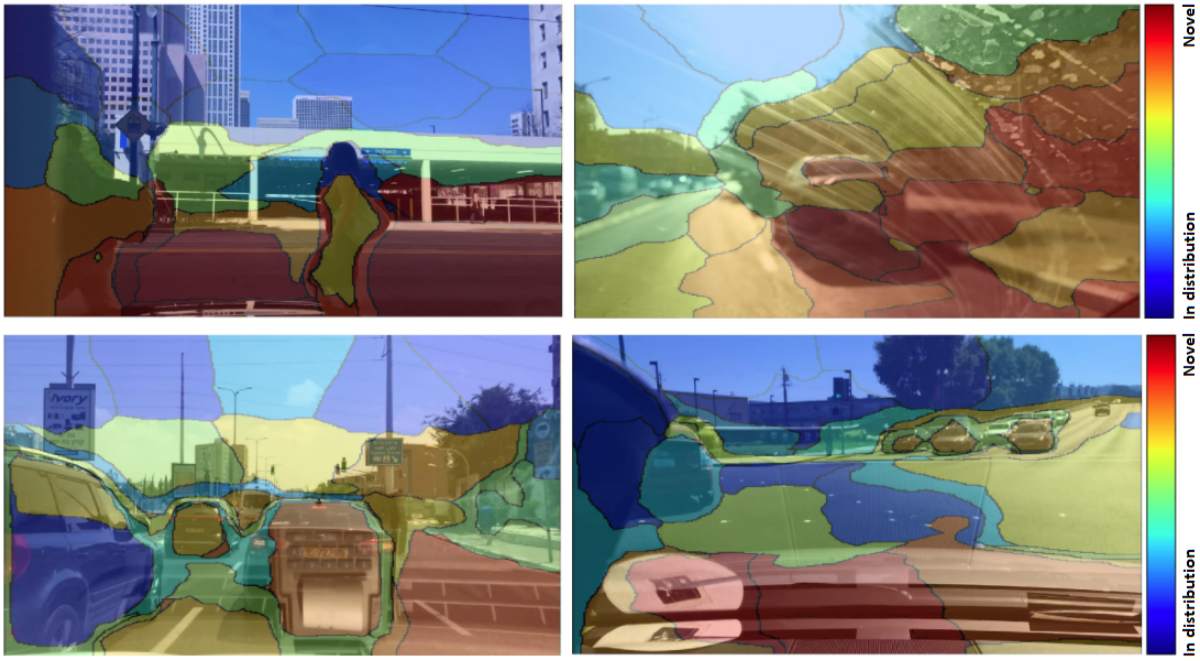

Fig. 7. Novel segments highlighted with a red to yellow color, and in distribution segments with a light blue to dark blue color, representing the likelihood of being drawn from the training distribution. In the upper left image pedestrians close to the car are highlighted as novelties, while the sky, buildings and other commonly appearing surroundings are considered in distribution. In the upper right image a very dirty windshield is included, marking most of the image as novel apart from the sky due to the occlusion. In the lower left image there is an reflection of a taxi receipt printer considered novel, with a similar scenario in the final image where a sign is reflected onto the hood of the car.

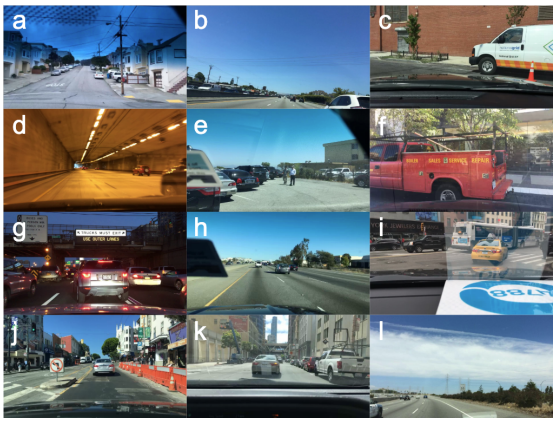

Fig. 8. Outliers the method ranks as having low probability of belonging to the training data. Many contain a miscoloration (a), misalignment (b), occlusions (a, e), blur (d,a) or a combination of them. Others without these defects miss visible road (c,f), show rare road signs (g), contain road works (j), include construction equipment/crane (k), have obstructions/reflections (h,i) or show cloud shapes, rare in this dataset (l).

1) Unsupervised Segmentation: Rather than relying on human interpretations, the method can be extended to work on segments of images, with the aim of finding the particular parts of the images contributing to the novelty. Following the methodology of [44] this process can be divided into three parts, first using a CNN to construct a pixel-wise embedding of an image, second a clustering method to partition these pixel embeddings into a segmentation and finally and third applying metric learning for clustering of the segments.

In other words, $f_\theta : \mathcal{X} \mapsto \mathbb{R}^{d\times N}$, where N is the number of pixels of $x \in \mathcal{X}$. Similar to how the distribution of features of all images $\mathcal{V}$ could be estimated by a von Mises-Fisher distribution, the goal is now to represent the distribution of pixels by a mixture of k such distributions with uniform prior,

$$F(v|\Theta) = \sum_{s=1}^{k} \frac{1}{k} f_s(v|\mu_s, \kappa) \tag{8}$$

where f_s is given by (7) with parameters μ_s and k, and $\Theta = \{\mu_1, ..., \mu_k, \kappa\}$. Again, let $\mathcal{V} = \{v_1, ..., v_N\}$ be the set of embeddings, but this time for each of the pixels and let $\mathcal{Z} = \{z_1, ..., z_N\}$ where $z_i = s$ denotes if pixel embedding v_i (i.e. the i:th pixel) belongs to segment s. The aim is then to maximize the log-likelihood given by,

$$\log P(\mathcal{V}, \mathcal{Z}|\Theta) = \sum_i \log \frac{1}{k} f_{z_i}(v_i|\mu_{z_i}, \kappa), \tag{9}$$

which is done through expectation maximization.

Finally the segment sorting is done by defining a prototype representing each segment. The natural selection of such vector is the mean direction vector μ_s of all embeddings within a segment. The network can now be trained in a similar way as for the image embeddings in the original method. Each pixel embedding v_i can be compared to the segment prototypes μ_s through cosine similarity.

2) Novel Segments: Four test examples of running the unsupervised segmentation on BDD100k, pre-trained on Pascal

VOC2012 [53], as in [44]; can be found in figure 7. With the help of this approach segments in the test images considered to be novel, compared to segments in the training data, can be highlighted. Even if the examples in figure 7 contain human interpretations, this approach may help with better understanding the context of a novelty in an automated setting.

C. Limitations

As discussed before, previous studies such as [17] have shown how novelty estimation can be used to handle uncertainties in predictions by choosing conservative driving actions. Although the third experiment shows how the suggested method can be used as performance prediction, no investigations are made into how to relate the magnitude of the estimation to physical driving actions. Such systems may also be able to collect data when encountering novel environments to retrain for similar scenarios in the future. The added benefit of using metric learning is the access of a metric which can relate the novel scenarios to the training data, for example indicating degree of sparseness. This can be applied to already existing datasets for use cases such as retrieving instances of a particular types of scenarios, by querying an example.

By using unsupervised learning, novelty detection can be used as a monitoring function independent of the underlying systems. However, as some studies have shown [30] [36], using knowledge of the underlying driving tasks, i.e. applying supervised or semi-supervised methods, additional information can be extracted for increased performance, at the cost of generality. Such systems might be able to run in parallel but are left out of this study.

VI. Conclusion

This paper introduced a novelty estimation algorithm based on unsupervised feature learning and von Mises-Fisher clustering. The presented experiments show promising results compared to current state of the art algorithms for novelty detection on the benchmarking datasets CIFAR-10 and CIFAR-100. In particular the method is applicable on real world driving data, where novel driving scenes could be distinguished with variable performance depending on the visual differences in relation to the training data. An example of how the novelty estimation could be used as a general monitoring system is given, where performance degradation in a segmentation network could be predicted. Finally, qualitative examples of novel segments in images were given to indicate a potential future use case.

The suggested method expands upon previous research regarding safe autonomous driving in general and monitoring of machine learning algorithms in particular. Although it is not a complete solution to the safety challenges it provides an additional layer of safety for machine learning based driving functions.

References

[1] L. Lubbe, H. Jeppsson, A. Ranjbar, J. Fredriksson, J. Bärgman, and M. Östling, "Predicted road traffic fatalities in germany: The potential and limitations of vehicle safety technologies from passive safety to highly automated driving," in *Proceedings of IRCOBI conference. Athena, Greece*, 2018.

[2] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[3] D. Åsljung, J. Nilsson, and J. Fredriksson, "Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 4, pp. 288–297, 2017.

[4] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3d object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, p. 1110, 2020.

[5] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[6] H. Kim, J. Cho, D. Kim, and K. Huh, "Intervention minimized semi-autonomous control using decoupled model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 618–623.

[7] A. Arıkan, A. Kayaduman, S. Polat, Y. Şimşek, İ. C. Dikmen, H. G. Bakır, T. Karadağ, and T. Abbasov, "Control method simulation and application for autonomous vehicles," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. IEEE, 2018, pp. 1–4.

[8] I. Batkovic, M. Zanon, M. Ali, and P. Falcone, "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users," in *2019 18th European Control Conference (ECC)*, 2019, pp. 256–262.

[9] M. Gyllenhammar, C. Zandén, and M. Törngren, "Defining Fundamental Vehicle Actions for the Development of Automated Driving Systems," in *SAE Technical Papers*, vol. 2020-April, no. April, 2020, pp. 1–10.

[10] S. M. Loos, D. Witmer, P. Steenkiste, and A. Platzer, "Efficiency analysis of formally verified adaptive cruise controllers," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1565–1570.

[11] 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.

[12] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal specification and verification of autonomous robotic systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–41, 2019.

[13] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[14] A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel, "Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3240–3247.

[15] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[16] S. Kojchev, E. Klintberg, and J. Fredriksson, "A safety monitoring concept for fully automated driving," in *2020 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2020.

[17] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," 2017.

[18] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[19] A. Ranjbar, C.-H. Yeh, S. Hornauer, S. Yu, and C.-Y. Chan, "Scene novelty prediction from unsupervised discriminative feature learning," in *2020 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2020.

[20] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[21] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, vol. 2, no. 5, p. 6, 2018.

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[23] D. J. MacKay, "Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks," *Network: computation in neural systems*, vol. 6, no. 3, pp. 469–505, 1995.

[24] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *arXiv preprint arXiv:1612.01474*, 2016.

[25] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," *arXiv preprint arXiv:1412.6596*, 2014.

[26] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[27] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1885–1894.

[28] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[29] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[30] D. A. Pomerleau, "Input reconstruction reliability estimation," in *Advances in neural information processing systems*, 1993, pp. 279–286.

[31] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Advances in neural information processing systems*, 2015, pp. 2791–2799.

[32] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 90–98.

[33] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[34] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[35] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 481–490.

[36] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 568–575.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[38] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[39] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," *arXiv preprint arXiv:1809.04758*, 2018.

[40] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[41] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[42] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[44] J.-J. Hwang, S. X. Yu, J. Shi, M. D. Collins, T.-J. Yang, X. Zhang, and L.-C. Chen, "Segsort: Segmentation by discriminative sorting of segments," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7334–7344.

[45] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 297–304.

[46] S. Hornauer, B. Yellapragada, A. Ranjbar, and S. Yu, "Driving scene retrieval by example from large-scale data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 25–28.

[47] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2174–2182.

[48] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[49] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Image anomaly detection with generative adversarial networks," in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2018, pp. 3–17.

[50] L. Wolf, S. Benaim, and T. Galanti, "Unsupervised learning of the set of local maxima," *arXiv preprint arXiv:2001.05026*, 2020.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[52] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, and G. Ridgeway, "Clustering on the unit hypersphere using von mises-fisher distributions." *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[53] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.