Scene Novelty Prediction from Unsupervised Discriminative Feature Learning

Arian Ranjbar[1*], Chun-Hsiao Yeh[2*], Sascha Hornauer[1,2], Stella X. Yu[1,2], Ching-Yao Chan[1]

***Abstract*— Deep learning approaches are widely explored in safety-critical autonomous driving systems on various tasks. Network models, trained on big data, map input to probable prediction results. However, it is unclear how to get a measure of confidence on this prediction at the test time.**

Our approach to gain this additional information is to estimate how similar test data is to the training data that the model was trained on. We map training instances onto a feature space that is the most discriminative among them. We then model the entire training set as a Gaussian distribution in that feature space. The novelty of the test data is characterized by its low probability of being in that distribution, or equivalently a large Mahalanobis distance in the feature space.

Our distance metric in the discriminative feature space achieves a better novelty prediction performance than the state-of-the-art methods on most classes in CIFAR-10 and ImageNet. Using semantic segmentation as a proxy task often needed for autonomous driving, we show that our unsupervised novelty prediction correlates with the performance of a segmentation network trained on full pixel-wise annotations. These experimental results demonstrate potential applications of our method upon identifying scene familiarity and quantifying the confidence in autonomous driving actions.

I. Introduction

Many complex deep learning methods, developed on image-based tasks, are applied nowadays on camera data in autonomous driving. Increasingly high task performance is achieved by using neural networks on videos e.g. for object detection or image segmentation [1]. However, in this domain, additional information about the expected performance of a neural network is needed to satisfy high safety requirements [2]. Generalization, robustness and reliability in new situations is often judged by testing on more or augmented data [3]. However, collection and annotation of test data is costly and can never cover all possible configurations of the world. One solution is to determine performance indicators while driving, so that control can be handed over to the driver if the expected performance of the network model degrades. After training on fixed training data, neural network based approaches can give high numerical activations for their chosen output in a new environment, but this does not necessarily correlate with task performance in the understanding of the designer of the task [4]. For example, a detected pedestrian, mirrored in the reflection of a car window can lead to high recognition scores when the network did not learn to associate the context of the scene.

[1]University of California, Berkeley, USA. {arian.ranjbar, saschaho, stellayu, cychan} @berkeley.edu
[2]International Computer Science Institute, Berkeley, USA. danielyehh@gmail.com
*These authors contributed equally to this work.

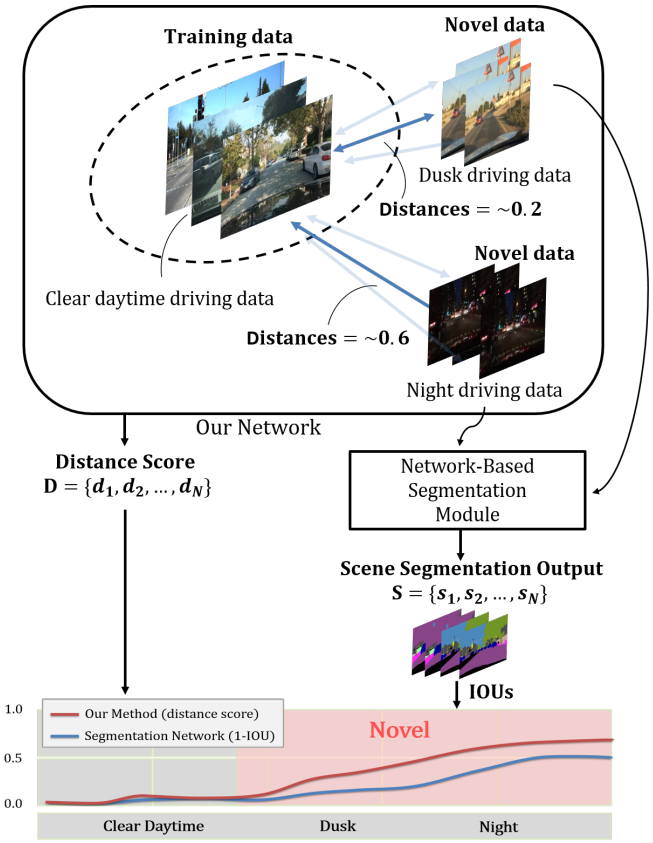

Fig. 1: Our proposed unsupervised network learns a similarity metric across all instances and calculates the similarity of novel driving input data compared to a fixed set of training data. To quantify the impact of novelty detection, the method is applied to a segmentation task showing it can approximate the performance of the segmentation network.

Recognizing this limitation, extensive work aims to classify unfamiliar environments as anomalies, compared to data considered normal. Successful detection of anomalies, used as input to a network, provides information on its expected performance, as it gives insight into its decision basis.

We approach the task of predicting a network's performance by quantifying the degree of similarity of novel data. In detail, we present a framework to calculate the similarity of novel camera input data compared to a fixed set of training data. This is done by using unsupervised feature learning to encode the input into a feature space in such a way that visually similar camera images are mapped close to each other. In the second step, we calculate the Mahalanobis distance from the feature vector of new input

to the distribution of our large-scale training data efficiently and in real-time.

To measure the general performance of the approach, we compare against other state-of-the-art work in anomaly detection. For a comparison within the autonomous driving domain, we show how our approach distinguishes camera data by appearance better than a baseline [5] which uses an autoencoder. Finally, we investigate how well we can approximate the performance of networks by comparing results on a segmentation task. The basic idea of our proposed method is illustrated in Figure 1.

In summary, our contributions are:

- A framework to estimate the similarity of new driving data to fixed training data, based on anomaly detection.
- Competitive performance on the aspect of anomaly detection with other state-of-the-art methods.
- Our approach scales to large video datasets for autonomous driving and can distinguish different environment conditions in data.
- We investigate how our network approximates the expected performance of SegNet [6] on a segmentation task.

II. Related Work

In the following, we first review more traditional methods to detect unfamiliar environments using anomaly detection. Then, we review previous literature on metric learning before introducing the main research this paper builds upon.

Anomaly Detection. Encountering unfamiliar environments has always been an issue when using neural networks for autonomous driving as the performance of the model degrades substantially. A major current research direction is unsupervised anomaly detection to identify these situations. Most methods fall into one of three approaches – conventional algorithms, autoencoders, and more recently, metric learning. While all three are able to isolate the anomaly, they have shortcomings in complex environments that prevents application in real world scenarios.

Conventional Methods. Methods such as One-Class SVM [7] and Isolation Forest [8] often show low performance in high dimensional data rich cases. To rectify this, conventional methods often require additional feature engineering such as dimensionality reduction to improve performance. They also require knowledge about the complete dataset before being able to identify anomalies, unapplicable to real world driving where novel scenarios may be encountered constantly.

Autoencoders. Autoencoders [9] have become the predominant approach to anomaly detection. These artificial neural networks learn a compressed representation of the data and try to minimize the reconstruction error [10], [5], [11]. Their performance differs on known and unknown data, thus making them a suitable proxy for anomaly detection. We compare against an approach used for safe robotic navigation [12] in Section IV-B. More advanced techniques, such as [13] use the compressed representation at the bottleneck to estimate a Gaussian mixture model. Autoencoders have a simple network structure, are easy to implement and remain computationally efficient with decent performance on smaller datasets. However, they do not scale well to large scale, high-res driving data where the performance degrades and exploitable differences between mapped anomalies and training data dissapear. Similarly, AnoGAN [14] used a GAN to generate samples of training data, and search for a point in the generator's feature space, which can generate a sample corresponding to given test sample. The reconstruction error is applied to define an anomaly score. AD-GAN [15] is also a GAN-based network that relies on an assumption; points that are badly represented in the latent space of the generator are likely to be anomalous. These approaches have difficulties distinguishing the training distribution and anomalies via the reconstruction error as crucial information may already be lost during the deconvolution layers or max pooling in the network.

Recent research has focused on the anomaly detection problem without reconstructing the data. Deep SVDD [16] is a network whose weights are optimized by a loss resembling the SVDD objective. [17] proposed an unsupervised learning scheme via utilizing local maxima as an indicator function.

Metric Learning. While the anomaly detection methods above can be used on simpler datasets, they are not suitable for complex real world driving environments. As an alternative, we examine several metric learning methods in this work. Metric learning aims to map data into a feature space, structured to minimize distance between similar instances. It has been widely used in many tasks such as face recognition [18] and image retrieval [19]. Other works use hand-crafted features with clustering methods [20]. However, this approach is not effective as the number of inputs grow and the dimensions increase. A number of works [21], [22] sample patches from images and yield the patches as supervision.

This paper mainly builds on research done by [23] on unsupervised instance level discrimination using a non-parametric softmax. Instead of classifying input images into certain classes, this method treats each instance image as its own class and the classifier is trained to identify each individual class. During training, the model will then learn a similarity metric across all instances and group visually similar instances closer together. This approach does not rely on labels which enables application on real world datasets without time-consuming labeling. It scales well to more data and deeper networks by using noise-constrastive estimation (NCE) to handle the computation cost, other approaches struggled with.

III. Novelty Prediction Method

Consider a training dataset $X = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$, for some input space of images $\mathcal{X}$[1]. The objective is to quantify the similarity between a new input data point $x \in \mathcal{X}$ and the training data X. For a low similarity score, such data points could be considered novel. This information can then

[1]The input space could in theory contain any kind of data but this work will focus on images in particular.

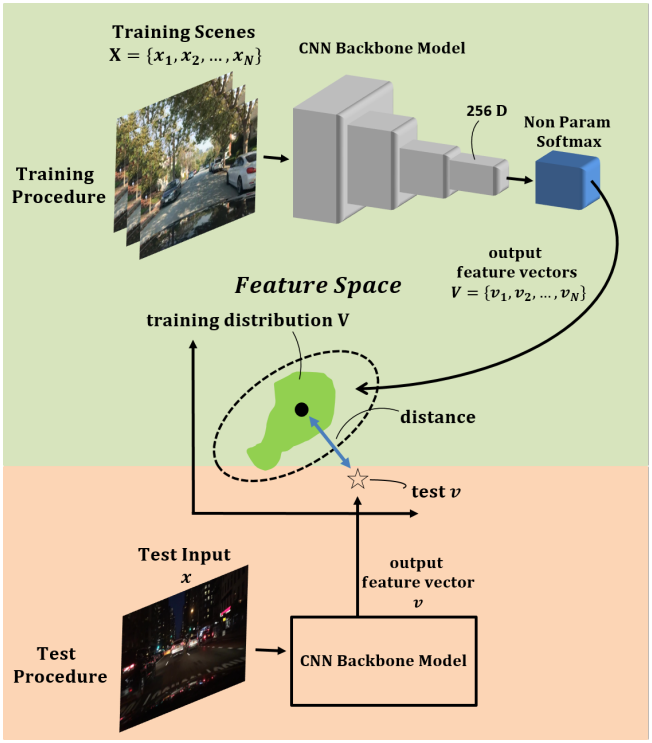

Fig. 2: The proposed unsupervised framework. We train the CNN backbone model which encodes images into 256-dimensional feature vectors. We then model the training data distribution in the feature space as a unimodal gaussian. Finally, we estimate the distance of individually mapped test images in the feature space.

be used to predict whether other functions using the same input data operate within an environment similar to which they were trained on.

Traditionally, such problems can be solved by providing labeled out-of-distribution data, which reduces it down to a classification problem. However, for tasks like autonomous driving, such data is not easily generated. This problem can instead be solved by learning a feature embedding function $f : \mathcal{X} \mapsto \mathbb{R}^d$, mapping images to a feature space of dimension d. The aim is to construct the feature embedding in such a way that similar images end up close to each other. A distance measure can then be introduced as an estimate of similarity between new data points and the training distribution.

A. Unsupervised Feature Learning

The feature embedding is constructed from a convolutional neural network, f_θ parameterized by θ. To achieve the desired property of having similar images close to each other, we adopt instance-level discrimination to train the network, based on previous work by [23]. Each image in the training dataset X is considered to be a distinct class and the feature outputs of the network are used to differentiate between each image instance.

The model is trained using a non-parametric softmax, rather than the more traditional parametric version, on the output features. The probability of an image x belonging to the i:th class is then given by

$$P(i|v) = \frac{\exp\left(f_\theta(x_i) f_\theta(x)/\tau\right)}{\sum_{j=1}^{n} \exp\left(f_\theta(x_j) f_\theta(x)/\tau\right)} \tag{1}$$

where τ is the parameter to control the density of the data distribution. The learning objective is then simply given by minimizing the log-likelihood,

$$\arg\min_\theta - \sum_{i=1}^{n} \log P(i|f_\theta(x_i)). \tag{2}$$

B. Distance Estimation

Using the feature embedding, the distance between two points in the feature space can be used as a metric of similarity. We can then measure the distance between a new data point and the training distribution by calculating the Mahalanobis distance. This is done by first mapping all of the training data X to features $V = \{v_1, v_2, \ldots, v_n\}$ and then calculating the empirical mean $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} f_\theta(x_i)$ and covariance $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(f_\theta(x_i) - \hat{\mu})(f_\theta(x_i) - \hat{\mu})^T$. The Mahalanobis distance between the new data point x and the training distribution is then given by,

$$M(x) = (f_\theta(x) - \hat{\mu})^T \hat{\Sigma}^{-1} (f_\theta(x) - \hat{\mu}) \tag{3}$$

Finally this distance is used as a measurement of novelty of the input data, since a greater distance would imply less similarity. This is further illustrated in Figure 2 where a schematic of the full pipeline is given.

Anomaly detection. When validation data is available, the accuracy can be further increased by extracting additional features from intermediate layers. This is especially useful in classification tasks, as shown in Section IV-A. In this case, a collection of features is gathered V_1, V_2, ..., V_N where $V_l = \{f_{l,\theta}(x_1), f_{l,\theta}(x_2), \ldots, f_{l,\theta}(x_L)\}$ and $f_\theta = f_{l,\theta} \circ f_{l,\theta} \circ \ldots \circ f_{l,\theta}$, for the training data X. The Mahalanobis distance is then calculated as above but for each layer M_l, i.e. between the set of intermediate feature V_l of the training data and the intermediate feature $v_l = f_{l,\theta}$ of the input data. Applying a similar strategy as in [24], the validation data is used to tune a weighted average $M = \sum_{l=1}^{L} w_l M_l$, where each weight w_l is gained through a logistic regression model. The full similarity estimation is illustrated in Algorithm 1.

IV. Experiments

In the following, we show three experiments to demonstrate the proposed method. In the first experiment, we compare our method against commonly used algorithms for anomaly detection. For the second experiment, the same method is tested on training data sets containing real world driving scenes in order to estimate similarity in new environments. Finally, we compare the performance profile of a segmentation network with our distance calculations.

Algorithm 1: Distance estimation

input : Training data $X = \{x_1, x_2, \ldots, x_n\}$, feature embedding f_θ, input image x, weights w_l

output: M

Initialize $[M_1, M_2, \ldots, M_L]$;

for $l = 1, ..., L$ **do**

  Calculate empirical mean $\hat{\mu}_l = \frac{1}{n}\sum_{i=1}^{n} f_l(x_i)$;

  Calculate empirical covariance $\hat{\Sigma}_l = \frac{1}{n}\sum_{i=1}^{n}(f_l(x_i) - \hat{\mu})(f_l(x_i) - \hat{\mu})^T$;

  Calculate Mahalanobis distance $M_l(x) = (f_l(x) - \hat{\mu})\hat{\Sigma}^{-1}(f_l(x) - \hat{\mu})$;

Estimate similarity as the weighted average $M = \sum_{l=1}^{L} w_l M_l$

A. One Class Novelty Prediction

From CIFAR-10 [25], one class was considered as in distribution and the rest as anomalies. The feature learning was done on data from one class and the corresponding Gaussian distribution was estimated in the feature space. Eight baselines were used in comparison.

Baseline Methods. We compare our method to state-of-the-art deep learning approaches and a few classic methods.

- **One-Class SVM [7] -** The one-class support vector machine (OC-SVM) learns a closed set in the input space, where the objective function learns to find a maximum margin hyperplane in the feature space that separates the mapped data from the origin. Samples which are located outside of the closed set are regarded as anomalies. The OC-SVM hyperparameters (the inverse length scale γ and ν) are adopted from the original paper in order to report the best result.
- **Deep Convolutional AutoEncoder [5] -** Deep autoencoders are neural networks using convolutional layers as the backbone architecture which learn a distribution of samples with an encoder that outputs a representation of reduced dimension. A decoder is employed symmetrically in the network structure in order to reconstruct samples accurately.
- **AnoGAN [14] -** In this method, one first trains a GAN-based network which generates samples according to the training data. Given a test sample, AnoGAN then finds a point in the generator's latent space which can generate a sample that is similar to the given test sample. An anomaly score is defined via the reconstruction error of sample. We incorporate the same architecture used by the original paper in our experiment.
- **DAGMM [13] -** This is an autoencoder-based approach that uses a Gaussian mixture model to perform density estimation on the representation of training samples generated by the autoencoder. Note that the architecture of the autoencoder is the same as the one in Deep Conv AutoEncoder [5] but with linear activation in the representation layer.
- **AD-GAN [15] -** Using a GAN-based model, this network learns a mapping from a low-dimensional normal distribution to the training data distribution. Given a test sample, the model estimates the inverse mapping from the image to low-dimensional feature, which is then used to generate a sample to compare with the original given one. Our experiment adopts the same DCGAN architecture [26] used in the original paper.
- **One-Class Deep SVDD [16] -** The one-class deep SVDD uses an objective similar to classic SVDD to learn a network while minimizing the volume of a hypersphere that maps the training data samples to the center of the sphere.
- **Local Maxima [17] -** This approach assumes the local maxima of each data point for some unknown value function. It then trains an indicator function and a comparator function, comparing the maximum values of the value function, in parallel to achieve an unsupervised one-class classifier for anomaly detection.
- **Latent Space Autoregression [27] -** This framework is composed of a deep autoencoder with a parametric density estimator. It uses the autoregressive procedure to learn the probability distribution underlying its latent representations. We follow the original paper by using ResNet-50 [28] as the backbone, either with an pre-trained model on Imagenet or CIFAR-10.

Experimental Protocol. We follow a similar experiment design used by [16] in deep anomaly detection using one-class classification benchmarks. For each experiment, we compare a set of images against the total dataset to detect the anomaly. The dataset is split into N classes for a total of N experiments. At each experiment, we assign a subset of class $N(n \subseteq N)$ to be the class of normal images, without any anomalies, in which the model builds a Gaussian distribution. The distance is then computed for images in the test set (all N classes), both normal and anomalous, which can then be used to evaluate the model's performance. Instead of determining the appropriate threshold to detect an anomaly, we use the area under the ROC (AUC) curve metric to evaluate the performance of the models. Note, however, that we need full knowledge of the ground truth labels of the test set in order to compute the AUC curve.

Hyperparameters and Optimization Functions. We adopt ResNet-18 [28] as backbone network and encode the output as 256-dimensional vectors in all of our experiments. We train using SGD with momentum 0.9 with batch size of 32 and set the weight decay hyperparameter to 4×10^{-5}. The learning rate is initialized to 0.01 and dropped by a factor 0.1 every 30 epochs after 80 epochs. We found the best performance already at epoch 90 after only one drop in learning rate.

Results on CIFAR-10. In Table I, ten experiments are shown using CIFAR-10. The first row contains results where the normal class is *airplane* (class 0 in CIFAR-10), and the anomalous instances are images from all other classes in CIFAR-10 (classes 1-9). In each row, we can see the average AUC results (over 10 runs) for all baseline methods. The

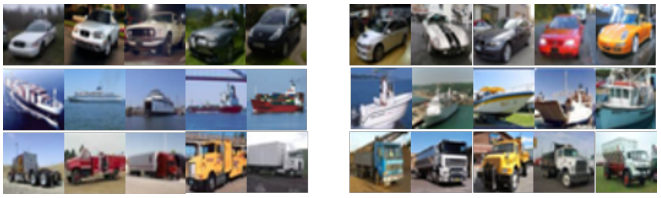

Fig. 3: Examples of most normal in-class samples determined by Local Maxima (left) and our method (right) for some CIFAR-10 classes (*automobile*, *ship*, *truck*) where Local Maxima performs the best.

results of our method and the percentage gain against the baseline methods are shown in the rightmost column. Our method shows a significant increase in performance against both conventional and deep learning state-of-the-art methods. In the case of *bird* (class 2 in CIFAR-10) and *cat* (class 3 in CIFAR-10), we obtain the best performance amongst the compared methods with 79.0% and 74.5% AUC. We outperform the most recent method, LSA [27], by 10.0% and 20.3% AUC.

For the classes *automobile, ship, and truck* (class 1, 8, 9 in CIFAR-10) we note that Local Maxima [17] performs better than our method. Figure 3 shows examples of most normal in-class samples according to Local Maxima [17] and our approach respectively. We extract the top examples using the highest probability of prediction to use during evaluation as the normal class. In examples where Local Maxima performs the best, especially in examples of *ship* (class 8 in CIFAR-10), the images all seem to have a similar global structure in the background (i.e. examples of *ship* all have blue sky or ocean as background).

Results on CIFAR-100. CIFAR-100 [25] represents a more significant challenge, as given by the low performances of most models, which is potentially due to the visual clutter between the large amount of classes. In each experiment, the 100 classes in CIFAR-100 are grouped into 20 super-classes, and the model is trained on the single "normal" class (i.e. class 0), and tested against all other classes (i.e. class 1-19). Our method outperforms other baselines and improves the AUC by 10% on average. Results can be found in the supplementary material.

Results on ImageNet-20. In addition to CIFAR, results are given for ImageNet-20. Images from ImageNet are grouped into 20 semantic concepts as in [29], where each concept has around 2800 images. The experiments are then carried out in the same way as the two previous setups. Our method beats autoencoders with 7% and current state of the art method CoRA [29] by 1.3%.

B. Driving Scene Novelty Detection

In the second experiment, we use the distance estimation as a measure of similarity to compare the training data to different driving scenes. Rather than using a fixed threshold for a binary classifier, it is defined as a continuous measurement on the novelty of input data. A subset of the images are picked out for training, such as images collected during the daytime with clear weather. The rest of the data containing

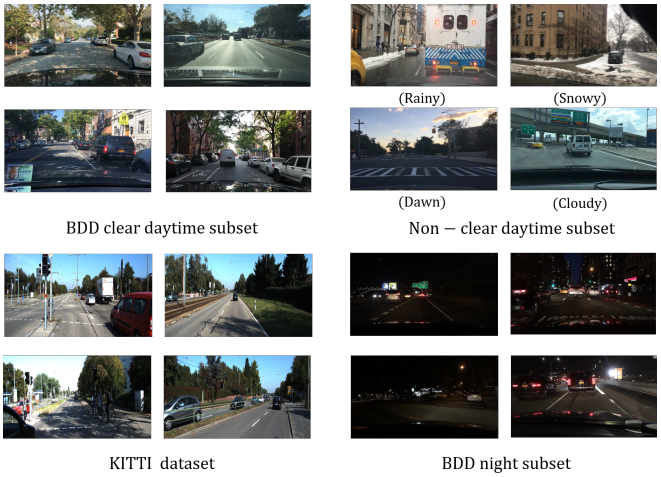

Fig. 4: Examples of driving scenes from each of the subsets of BDD100k and KITTI under daytime, night and different weather conditions.

various weather scenes can then be used for evaluation, to show that we can identify novel scenes.

Driving Datasets. We consider two common driving datasets in our experiments: KITTI [30], and BDD100k [31], which are described below. Note that all images are resized to 224 x 224, and the pixel values are normalized to [-1, 1].

- **KITTI:** The KITTI dataset is an outdoor dataset acquired entirely in the city of Karlsruhe, Germany. The original image size is 1,242 x 376. The sub-dataset in KITTI meant for object tracking is used. It consists of 8,008 training and 7,518 testing frames. Some examples of images from KITTI can be found in Figure 4.
- **BDD100k:** The BDD100k benchmark has 100k images (1,280 x 720 px) and is collected in the United States, mostly in San Francisco and New York, under a wide variety of driving circumstances. Attributes include time of day, weather conditions and location (highway, city, parking lot, etc.). The full dataset contains 70,000 training, 20,000 test, and 10,000 validation images. Examples of BDD100k can be found in Figure 4.

Experiment Setup. Previously, autoencoders have been suggested for novelty detection in simple driving scenes [12]. We conduct the experiments in comparison to DCAE [5], using the originally proposed structure. This included three modules consisting of 128, 64 and 32 x (5 x 5 x 3)-filters in the encoder, where the decoder is created symmetrically while replacing max-pooling with upsampling. The model is trained for 250 epochs with MSE loss, using a batch size of 32 and a fixed weight decay hyperparameter of 10^{-6}.

The baseline is compared to our network which uses the same network structure as in Section IV-A. We divide the BDD100k dataset according to the attribute information in the annotation file [31]. In particular, we consider weather and the time of day attributes, further dividing the clear daytime subset into a training and testing set which can be seen in Table II. Training both our model and the DCAE on the clear daytime dataset, the models' performance is

	OC-SVM	DCAE	ANO-GAN	DAGMM	AD-GAN	DEEP SVDD	Local Maxima	LSA	Ours	Gain
airplane	61.6	59.1	67.1	41.4	64.9	61.7	74.0	73.5	**76.6**	**2.6%**
automobile	63.8	57.4	54.7	57.1	39.0	65.9	**74.7**	58.0	69.6	-
bird	50.0	48.9	52.9	53.8	65.2	50.8	62.8	69.0	**79.0**	**10.0%**
cat	55.9	58.4	54.5	51.2	48.1	59.1	57.2	54.2	**74.5**	**15.4%**
deer	66.0	54.0	65.1	52.2	73.5	60.9	67.8	**76.1**	71.9	-
dog	62.4	62.2	60.3	49.3	47.6	65.7	60.2	54.6	**72.0**	**6.3%**
frog	74.7	51.2	58.5	64.9	62.3	67.7	75.3	75.1	**77.9**	**2.6%**
horse	62.6	58.6	62.5	55.3	48.7	67.3	68.5	53.5	**70.3**	**1.8%**
ship	74.9	76.8	75.8	51.9	66.0	75.9	**78.1**	71.7	77.4	-
truck	75.9	67.3	66.5	54.2	37.8	73.1	**79.5**	54.8	76.9	-

TABLE I: Average AUCs in % (over 10 runs) of anomaly detection methods on CIFAR-10. Our method achieves the best performance against both conventional and deep learning baselines on most CIFAR-10 classes. We also show our gain in % compared against the best baseline.

BDD subset	# frames	Description
Clear Daytime	training: 11,000 test: 1,453	weather: clear time: daytime
Night	27,970	weather: all time: night
Non-clear Daytime	24,273	weather: cloudy rainy, foggy, snowy time: daytime

TABLE II: The subsets we used in the experiment. The number of frames and description of each subset are included.

evaluated on the ability to distinguish the novel environments from the testing data. For DCAE, this is done by using the reconstruction error of the images.

Results. Results are illustrated in Figure 5, where the similarity estimation is compared between the BDD clear daytime training dataset and each other subset. Our method distinguishes well between the training data and the novel environments. Furthermore, the KITTI data lies closer to the training data than the BDD non-clear daytime subset. This is because the KITTI data mainly consists of clear daytime environments which is more similar to the BDD clear daytime training data than the foggy, rainy and snowy images in the BDD non-clear daytime subset. Finally, the BDD night subset is the furthest distance away as it contains the lowest similarity to the clear daytime images. The same results can be seen for training on KITTI as well.

The DCAE model struggles to differentiate the datasets as seen in Figure 5. As [12] addresses, autoencoders may not work as well for highly varied datasets of unique images since the network needs to capture complicated structures in the data rather than learning pixel-copy representation. However, it still shows a marginal gap in reconstruction loss between the day and night datasets.

C. Segmentation Task Performance Prediction

As the final experiment, we illustrate how novelty detection can be used as confidence measure in an exemplary network for an autonomous vehicle pipeline, inspired by [12]. We investigate a basic segmentation network, SegNet [6] which is often applied in autonomous driving tasks [32], [33], and the reliability of its predictions based on the novelty of the testing data. Both networks are trained on the same training data and the similarity estimation of our model acts as a proxy for the performance of the segmentation network.

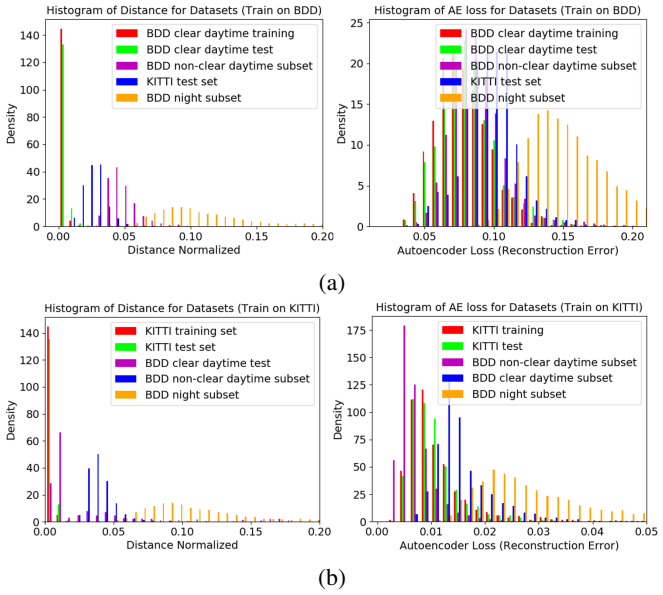

Fig. 5: Histogram of the distance (our method) and reconstruction loss (DCAE) based on training (a) BDD clear-daytime subsets, (b) KITTI training set, and test both (a) and (b) on BDD clear daytime test, BDD non-clear, KITTI test, and BDD night subsets. The results of (a) and (b) clearly show that our method differentiates several environments effectively, while our compared baseline, DCAE does not.

Experiment Setup. We show that segmentation loss can be modeled by the similarity of the input images compared to the training data. We use the BDD10k subset which contains segmentation labels. Of the 10,000 images only 2972 contain both a segmentation mask, weather and time information. Of these 829 have clear weather and were taken during the daytime. 729 of these were used for training and 100 were added to the rest of the images for testing.

Results. Results are found in Figure 6 where the scatterplot illustrate the similarity estimated from our network and segmentation loss from SegNet. Although SegNet tends to generalize well, a decrease in performance can be seen for test data of low similarity. In particular, this is evident for the night time images. Furthermore, the high variance may be due to some scenes having better lighting conditions to others, even during the night time images. Results for other conditions can be found in the supplementary material.

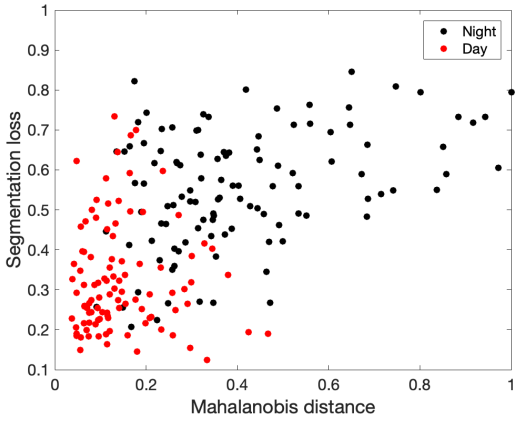

Fig. 6: Segmentation loss plotted against similarity for the 100 clear daytime testing images combined with 115 night-time testing images. Night time images both show higher segmentation loss and less similarity.

V. Summary

We introduce a framework to estimate the similarity of new data to a fixed training set. The method is compared to regular state-of-the-art anomaly detection algorithms where our method outperforms the majority of the classes on CIFAR-10, CIFAR-100 and ImageNet-20. Furthermore, we show that the method scales to large datasets for autonomous driving by applying it to the KITTI and BDD100k dataset, where we successfully estimate the similarity of driving scenes. For an autonomous vehicle, this can be used to predict whether it is driving in an environment similar to its training data. Finally, we show an example of such application where we predict the performance profile of a segmentation network as it encounters novel scenes. Our proposed approach can significantly contribute to the field of autonomous driving as it provides an indicator of expected performance of a driving module in novel environments.

References

[1] M. Wulfmeier, "On Machine Learning and Structure for Mobile Robots," pp. 1–25, 2018.

[2] 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 *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, (California), pp. 4745–4753, International Joint Conferences on Artificial Intelligence Organization, aug 2017.

[3] T. Dreossi, S. Ghosh, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Systematic Testing of Convolutional Neural Networks for Autonomous Driving," *Journal of the Brazilian Chemical Society*, vol. 9, pp. 219–223, aug 2017.

[4] A. Subramanya, S. Srinivas, and R. V. Babu, "Confidence estimation in Deep Neural networks via density modelling," 2017.

[5] A. Makhzani *et al.*, "Winner-take-all autoencoders," in *Advances in neural information processing systems*, pp. 2791–2799, 2015.

[6] 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.

[7] 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.

[8] F. T. Liu *et al.*, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, IEEE, 2008.

[9] G. E. Hinton *et al.*, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] 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*, pp. 90–98, SIAM, 2017.

[11] 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.

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

[13] B. Zong *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," 2018.

[14] T. Schlegl *et al.*, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*, pp. 146–157, Springer, 2017.

[15] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Anomaly detection with generative adversarial networks," 2018.

[16] L. Ruff *et al.*, "Deep one-class classification," in *International Conference on Machine Learning*, pp. 4390–4399, 2018.

[17] L. Wolf, S. Benaim, and T. Galanti, "Unsupervised learning of the set of local maxima," 2018.

[18] F. Schroff *et al.*, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[19] H. Tang and H. Liu, "A novel feature matching strategy for large scale image retrieval.," in *IJCAI*, pp. 2053–2059, 2016.

[20] D. Han and J. Kim, "Unsupervised simultaneous orthogonal basis clustering feature selection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5016–5023, 2015.

[21] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer, "Cliquecnn: Deep unsupervised exemplar learning," in *Advances in Neural Information Processing Systems*, pp. 3846–3854, 2016.

[22] A. Dosovitskiy *et al.*, "Discriminative unsupervised feature learning with convolutional neural networks," in *Advances in neural information processing systems*, pp. 766–774, 2014.

[23] Z. Wu *et al.*, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

[24] K. Lee *et al.*, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[26] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[27] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "And: Autoregressive novelty detectors," *arXiv preprint arXiv:1807.01653*, 2018.

[28] 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*, pp. 770–778, 2016.

[29] K. Tian *et al.*, "Learning competitive and discriminative reconstructions for anomaly detection," *arXiv preprint arXiv:1903.07058*, 2019.

[30] A. Geiger *et al.*, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[31] 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*, 2018.

[32] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2156, 2016.

[33] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, "A comparative study of real-time semantic segmentation for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 587–597, 2018.