2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Task-Aware Novelty Detection for Visual-based Deep Learning in Autonomous Systems

Valerie Chen, Man-Ki Yoon, and Zhong Shao
Department of Computer Science, Yale University
Email:{v.chen, man-ki.yoon, zhong.shao}@yale.edu

***Abstract*— Deep-learning driven safety-critical autonomous systems, such as self-driving cars, must be able to detect situations where its trained model is not able to make a trustworthy prediction. This ability to determine the novelty of a new input with respect to a trained model is critical for such systems because novel inputs due to changes in the environment, adversarial attacks, or even unintentional noise can potentially lead to erroneous, perhaps life-threatening decisions. This paper proposes a learning framework that leverages information learned by the prediction model in a task-aware manner to detect novel scenarios. We use network saliency to provide the learning architecture with knowledge of the input areas that are most relevant to the decision-making and learn an association between the saliency map and the predicted output to determine the novelty of the input. We demonstrate the efficacy of this method through experiments on real-world driving datasets as well as through driving scenarios in our in-house indoor driving environment where the novel image can be sampled from another similar driving dataset with similar features or from adversarial attacked images from the training dataset. We find that our method is able to systematically detect novel inputs and quantify the deviation from the target prediction through this task-aware approach.**

I. Introduction

One of the most successful examples at the forefront of autonomous systems is self-driving cars, vehicles with the ability to sense their environment and navigate the road without human direction and supervision. These technologies are powered by machine learning algorithms trained extensively on mass amounts of data collected from driving in real life and in simulation [1], [2]. As this technology rapidly progresses, there is an increasing concern with regard to safety. Deep neural networks are not trained with safety concerns in mind and are themselves a cause of worry due to the lack of transparency in its decision-making process. Trust in safety-critical autonomous systems like self-driving cars is tied directly to the amount of knowledge we have of the internal decision-making mechanism. It is non-trivial to determine what types of situations a model is able to make a safe decision and what types it will make an erroneous and perhaps life-threatening one. Recent works have shown that simple adversarial attacks such as the addition of noise can drastically change the prediction of the model [3], [4], [5]. These attacks need not be adversarial, as a shift in lighting or the addition of "rain" on an image can also affect the model prediction [6]. Detection of these attacks in today's world is an important features that we would ideally like to have in deep-learning models.

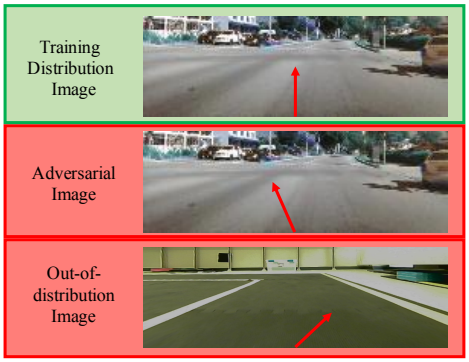

Fig. 1. (Top) The original image and correctly predicted steering angle of 0 degrees. (Middle) The Fast Gradient Sign Method (FGSM) attack image and predicted steering angle of 26 degrees at an $\epsilon = 0.001$ level. (Bottom) Out-of-distribution image. The middle and bottom examples demonstrate two classes of novel images we consider: adversarial and out-of-distribution.

Novelty detection is relevant when the quantity of "novel" data is insufficient to construct an explicit model of out-of-distribution classes, rather the general approach is to model the "in-class" data [7]. Novelty detection is a difficult problem because the amount of available "in-class" data is small relative to the amount of the possible "out-class" data. For example, a model trained on only the MNIST numbers [8] should be able to determine when it is presented with letters, rather than the numbers the model was trained on, that the letters are novel data. A more subtle "out-class" data is also the set of adversarially-modified of seemingly similar images of MNIST numbers that would actually distort the prediction module's accuracy.

For visual-based systems, novelty detection typically hinges on determining the novelty of the input image and its similarity to the large set of training data that the model has seen [9], [10]. These approaches can be overly pessimistic with held-out images from the same dataset that the model might still be able to make a good prediction on. On the other hand, these approaches can also be overly trusting of images that look close in similarity to the training set and yet would produce a poor prediction. This is because these methods are able to handle only environments where the images are highly structured. In real-world problems, the difficulty arises from the high dimensional, diverse space from which the images are sampled from. We would like to have models that can determine novelty not only for structured datasets, but also for real world autonomous systems. A robust novelty

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

detection method should be able to detect images not only from an entirely unseen environment but also from images of a seen environment that have been slightly modified through adversarial attacks.

In this paper, we show that novelty detection can be made *task-aware* when provided with key characteristics of the input data from which the task-specific decision is being made. For this, we present a simultaneous training framework for learning both the visual-based prediction model and novelty detector using network saliency extracted from the prediction task. We apply our methods to the prediction of steering angle given the image of the road ahead in autonomous-driving setting. We conduct experiments on open-source driving datasets collected *in the wild* as well as in our own *in-house* driving environment. While the proposed method is not able to provide concrete guarantees on *all* types of novel input, we show that it is able to detect a wider array of novel scenarios, when compared to prior methods for novelty detection. When considering specific applications with varying fault tolerance, our novelty detection mechanism gives a quantifiable result for how to trust a new piece of input. Then based on the specific applications, one may set thresholds or act accordingly based on the novelty scores. For example, in an autonomous-driving setting, the car could slow down when a highly novel situation is encountered.

II. Motivation

In this work, we are interested in analyzing a deep-learning model to determine when a novel scenario has occurred. This is relevant for safety-critical systems because we cannot simply trust the trained learning model without specific guarantees or empirical results. The method that we propose can be applied to a deep-learning model $M(d)$ trained on an input set D to detect novel inputs. Novelty can be characterized in two different ways: (i) out-of-distribution and (ii) adversarial perturbation. For the case of (i), a new input d' is from an out-of-distribution dataset D' that is dissimilar to the training set D. In the sense of (ii), an input $d' = d + \epsilon$ is considered to be novel when an image d, which is in-class with respect to the training set D, is perturbed by ϵ and changes the prediction so $M(d)$ significantly differs from $M(d')$. Specific examples of such adversarial perturbations are provided in later sections. To be task-aware, we desire that the network should not only be able to detect the first case but also the latter, which is arguably more imperative for safety-critical systems. For these systems, the novelty detector can help the system revert to safer, conservative decision-making or alert a human operator to take over, when a novel, untrustworthy, and unfamiliar situation is detected.

A. Related Work

Novelty detection approaches can be classified into a few different categories: traditional probabilistic and distance-based novelty detection, which often suffer from the curse of dimensionality, and newer reconstruction and domain-based novelty detection [7]. The latter is often preferred to model the underlying data in safety-critical systems.

Recent approaches to novelty detection have been focused on designing and training one-class classifiers. In an one-class classifier, all data points in the training set are considered within the target class and all other possible data points are considered novel, so the novel class is disproportionately large compared to the target class. These type of one-class approaches [11], [10] have been largely focused on classification applications. Experimental results have been performed on datasets including MNIST, CIFAR10 [12], and Caltech-256 [13], which are all highly structured and distinct object datasets. Similarly Hendrycks and Gimpel [14] and Liang *et al.* [15] consider settings only where the out-of-distribution data is clearly defined by a large corpus, making the assumption that we have an oracle for what out-of-distribution data could look like. In reality, for real world systems, it is difficult to quantify with a dataset what the out-distribution could be. For example, a road trained on sunny images could make erroneous predictions on cloudy images, making the visually similar cloudy images novel for the model [6].

In the context of robotic systems, Richter and Roy [9] has provided preliminary results for an approach to novelty detection for autonomous systems through training an autoencoder to learn a representation of the training data and determine novelty from the mean square reconstruction error of a new image. The authors note that even their environment is still highly structured and not representative of real driving environments. Hence, their method of utilizing an autoencoder to memorize training images produced good results. In a similar application as [9], more recent work by McAllister *et al.* [16] proposed the use of a generative model, a variational autoencoder (VAE), to handle robustness to out-of-distribution inputs. The approach transforms a new input using the VAE into the training input distribution for more robust prediction. While the goal is slightly tangential to ours, we believe that the VAE model is an interesting one to compare against for novelty detection.

We are interested not only in detecting novelty in terms of out-of-distribution data (i.e., a dataset sampled from an entirely different distribution), but also detecting adversarial perturbations that can bring about novelty that will directly affect the task even though the input image may not seem visually different from what can be seen from the training set (example shown in Figure 1). While adversarial attacks on categorical prediction variables with classes is frequently studied [17], the problem of adversarial attacks on a regression problem is still an open question [18]. One white-box adversarial attack that we consider in this work is the Fast Gradient Sign Method (FGSM) [19]. FGSM is a simple, yet effective L_∞ bounded attack where given training point (x, y), loss function J, model parameters θ, perturbation ϵ, a resulting adversarial input $\tilde{x}$ is computed as:

$$\tilde{x} = x + \epsilon \cdot \texttt{sign}(\nabla_x J(\theta, x, y))$$

White-box attacks [6] assume access to the model parameters θ, which is not necessarily realistic.

III. Task-aware Novelty Detection

In this paper, we target an application of predicting steering angle from the raw pixels of a given road image captured by a single front-facing camera [2], which is a regression problem. The prediction network from the road image is typically in the form of a Convolutional Neural Network (CNN) which allows the network to detect useful road features for steering prediction without the need for human-selected criteria.

A. Incorporating Task Awareness

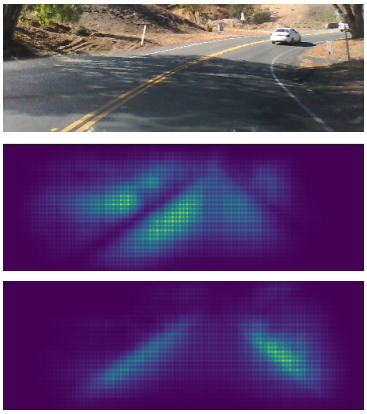

Fig. 2. A preliminary experiment on the Udacity self-driving car dataset to demonstrate that VBP masks are tied to learned features. (Top) Original image view of the road (Middle) Generated VBP mask on network trained with random steering angles (Bottom) Generated VBP mask on network trained with actual driving angles. This demonstrates that given meaningful input and output, VBP can extract key areas of an image such as the edge of the lane.

The first criteria of task-awareness is determining the key characteristics of the input image from which the decision is being made. This is often a challenge for deep learning models, which are often viewed as a black box system because simple inspection of numerical weights of the network do not convey its decision-making process. Network saliency visualization methods such as Visual Backpropagation (VBP) [20] give insight on what aspects of an input caused the output. In particular, as shown in Figure 2, VBP identifies sets of pixels of the input image that contribute most to the predictions made by a trained CNN through combining feature maps from deeper convolutional layers with more relevant information with higher resolution feature maps of shallow layers through a series of deconvolutions in a backward pass. We extend work by [21] to use network saliency for not only dimensionality reduction, but also for capturing the task-relevant features.

Additionally, a second key criteria to task-awareness is the accuracy of the prediction (e.g., on steering angle), that is the error $||y - p(x)||_1$, where y is the true value and $p(x)$ is the predicted output for input x. We consider absolute error because the magnitude is the aspect relevant to novelty rather than directionality. For example, a prediction off by +20 or -20 are both wrong. We claim that it is intrinsically tied to the network saliency and that an input that causes a distortion to the VBP image will cause also a distortion to the prediction accuracy. Hence, we propose to utilize this tie between network saliency and accuracy to detect novelty. Given the network saliency map for an input image and a representation of its prediction, we can infer whether the difference between that prediction and the actual label is close or not. For example, in the first type of novelty where the image is clearly out-of-distribution, the network saliency map would not look like any normal saliency map generated by input from the training distribution. In the second type of novelty where the input image is more similar to that of the training data, the noise or perturbation in the image that causes the network to make a potentially-incorrect prediction would be captured in the perturbed prediction representation that would not match those that have previously been seen as correct representations with similar saliency maps.

B. Training Process

We implement our approach by instantiating a two-network architecture, as shown in Figure 3. The first of which is the conventional predictive module [2] and the second of which is the network-saliency based novelty detector module. The predictive model takes in an image as input (60x160) and consists of convolutional layers (e.g., 5 layers with kernel sizes of 5x5/5x5/5x5/3x3/3x3) followed by fully connected layers (e.g., 3 layers with 100, 50, and 10 nodes, respectively) with ReLU activation into one singular output node for steering angle, trained with mean squared error loss. The novelty detection module takes in as input a VBP image (60x160) and vector of size 10 which captures the predicted steering angle. The VBP image is passed through a convolutional layer (kernel size 4x4), max-pooling, and ReLU activation into a fully connected layer (size 256). The set of VBP images are extracted through one backpropogation step, we extract the current network saliency based on the current state of the network using the input batch. The vector is passed through a fully connected layer with ReLU activation. The vector is extracted from the input to the last fully connected layer from the prediction network. The two inputs are concatenated and passed through a final fully connected layer with ReLU activation, again with a singular output node which is novelty, the absolute error between the predicted steering angle and actual steering angle. We define error prediction as $n(x) := ||y - p(x)||_1$, where x an input image, y the true value (but unknown for test image) and $p(x)$ the predicted output.

The novelty prediction forward-pass is then repeated with adversarial training. Using FGSM, we generate a set of adversarial images from the current training batch. We select an $\epsilon = 0.02$ value based empirically on selecting an ϵ value that consistently changes the output prediction. These adversarial images are first passed through the prediction module to extract the VBP images and prediction vector, which are then used as input as another forward pass to train the novelty detection module. We do not backpropogate the loss of the adversarial image for the prediction module since the goal here is not to train the prediction module

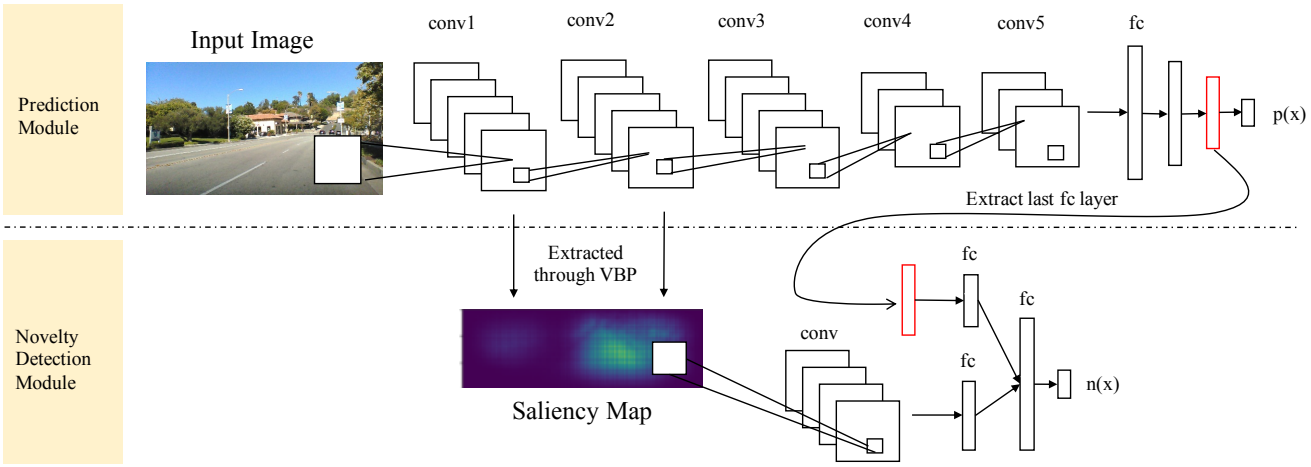

Fig. 3. The network architecture for our novelty detection method. The network on top is the prediction module which maps input image to steering angle using a standard CNN. The network below is the detection module which maps saliency map and prediction representation, extracted through a pass through the prediction module, to a novelty score. The training process consists of two passes through the whole architecture, once with normal input images and another with adversarially modified input images. However, only the first is backpropogated through the prediction module.

to be robust to adversarial modifications. Note that through these two forward passes the none of the gradients from the novelty detection module are backpropogated to the prediction module, so prediction accuracy would remain the same as in the standard training procedure.

In summary, the training process is as follows:

1) Iterate one step of the prediction module: $p(x)$
2) Backpropogate loss and generate the VBP mask
3) Extract last fully connected layer
4) Forward pass and backprop the detection module: $n_1(x)$
5) Fast Gradient Sign Method with $\epsilon = 0.02$ on batch.
6) Forward pass and backprop the detection module with attack image: $n_2(x)$

In the testing phase, we first predict the steering angle, $p(x)$, using the prediction module, then generate $n(x)$ from the novelty detection module. By the definition, the $n(x)$ value grows as the prediction module cannot produce a correct $p(x)$ for the input x. We can then compare this error to the distribution of the errors that we previously saw during training of the target class. We consider an input image to the architecture to be novel if the error is above a certain threshold that can be experimentally determined from the error distribution obtained in the training phase. We select the $\alpha = 0.01$ threshold based on [9]. This means that an input is considered novel when the novelty score is in the 99th percentile of the distribution of $n(x)$ where x from training set. In later sections, we demonstrate that through experimental results, the threshold was not necessary because the network learned a large separation between training and novel data, but could be utilized if needed.

IV. Experimental Results

A. Ablations to Task-Aware Network

We experimented with a few variations of the novelty detection module to quantify whether and which components of the proposed architecture are necessary to be able to determine novelty. The results presented in Figure 4 compare the proposed approach, Task-Aware, against the following three variations.

No VBP: The first ablation that we consider is the use of network saliency as part of the input to the novelty detector network. One can consider whether the use of the raw pixel image and the prediction representation is sufficient to also make decisions about how large the error would be. We verify that our results produce a distribution of novelty prediction scores that matches the reconstruction error loss distributions of [21].

No Adversarial Training: A second variant of the task-aware network that we consider is removing the adversarial training step in each iterations. With the inclusion of adversarial training, the model is able to see some examples of what incorrect behavior and perturbations to the input data may look like.

No Parallel Training: The third ablation we consider is removing the parallel training of the prediction network and the novelty detection network. In this set-up, we first train the prediction module, and then subsequently train the novelty detection module separately. We find that the model is not able to learn to separate the novel data as well because it was exposed to less variation of training data and prediction errors, which is a benefit of training both networks simultaneously (thus *task-aware*).

B. Novelty Detection

The main experimental question is whether incorporating a notion of task-awareness can detect novel scenarios that could be encountered by a driving model that would produce potentially dangerous results. We explore the benefits of the proposed simultaneous methods not only through several ablations as explained above, but also through comparisons against existing methods:

1) Generative networks such as Autoencoders (AE) and

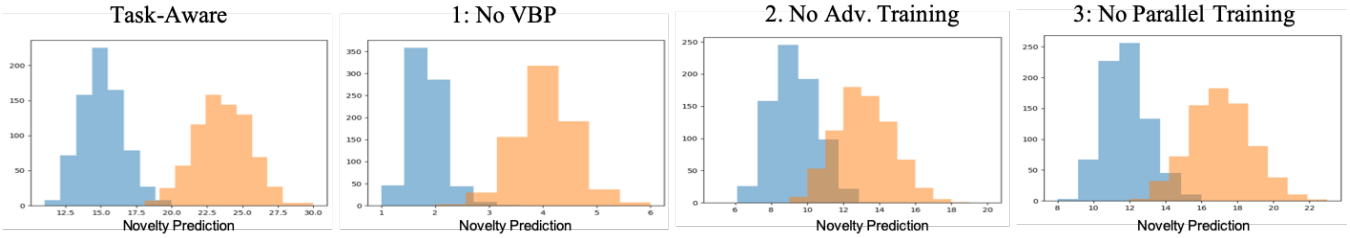

Fig. 4. Histogram comparison of predicted novelty scores ($n(x)$) for all ablations of our task-aware model, where the blue is the training data and orange is the attack data. A good novelty detector should be able to clearly separate the two distributions. We conduct these experiments specifically on adversarial attacks which are the most difficult to detect, since most of these variants performed equally on detecting out-of-distribution data as in the experiments from Table I. We find that the final Task-Aware set-up outperforms other variants. For varying set-ups, the absolute values of $n(x)$ will vary but we can still compare the different set-ups or methods by looking at the overlap between the blue and orange histograms.

Variational Autoencoders (VAE) [9], [16] that leverage reconstruction loss.

2) Modified generative networks using VBP, AE+VBP and VAE+VBP, as a form of feature extraction and dimensionality reduction [21].

Since these methods are based on reconstruction of the input image, to generate the VBP image for the modified generative network, we use the same prediction module that was trained for our task-aware to maintain the same generated network saliency maps. We train the autoencoders with standard mean squared error loss and the variational autoencoders with the summed loss of reconstruction loss with the Kullback-Leibler divergence loss. Novelty is detected by establishing a cut-off threshold in reconstruction loss based on the training loss distribution and determine if the reconstruction for a new image is greater than the cut-off. In our experiments we use a cut-off of $\alpha = 0.01$ as in [9].

C. Out-of-distribution Datasets

Fig. 5. Example training images from the three driving-based datasets that we considered each of which was collected with a camera mounted on the front of the car with the steering angle recorded. Two are collected *in the wild* and one indoors, but the middle and right have similar lane markings.

For our experiments, we work with three different autonomous driving datasets. The first of which is the Udacity self-driving car dataset [22], which consists of over 45,000 images collected from actual driving in Mountain View, CA. Additionally, we work with the Comma.ai dataset [23] and subset it to be about the same size as the Udacity dataset. We also collected an in-house dataset from a model car driving in an indoor driving environment. The roads in our model self-driving environment have varied surroundings and backgrounds, which provides for more variety than hallway-style environment studied in [9] and [16]. We notice that there are similarities that the Comma.ai dataset shares with both of the other datasets as described in Figure 5. Each dataset has annotated images with associated steering angles.

Table I presents the results of comparison between 3 different driving datasets where TRAINING is the dataset that the prediction module was trained on and TEST is the out-of-distribution dataset. A checkmark denotes that *all* of the images in the corresponding test set were classified as novel with respect to the training set, no checkmark denotes that *none* of the images were correctly classified as novel, and a number denotes the detection rate that is not 0 or 1. We found that for the most part either the learned representation was able to capture the distinctive features or it was not, which is why the test set was either categorized as all novel or all not novel. Out-of-distribution datasets, albeit that ours are similar in application, are still fundamentally different and therefore relatively easy to distinguish apart. We observe that our method is able to perform as well as the comparison methods. This set of experiments acts as a baseline comparison to demonstrate that we are able to detect the first type of novelty which is out-of-distribution data. One observation with regards to the generative methods is that if the test set is too similar or too simple compared to the training set, then the method struggles to detect novelty (e.g., AE for Udacity/In-House). For example, the landscape of our in-house data is significantly more plain than the Udacity dataset, which is a reason why the in-house data actually had a smaller reconstruction loss than the training set itself.

TABLE I

OUT-OF-DISTRIBUTION PAIRWISE COMPARISON.

Training / Test	AE	AE+VBP	VAE	VAE+VBP	OURS
Udacity / In-House		✓		✓	✓
Udacity / Comma.ai	✓	✓			✓
In-House / Udacity	✓		✓	0.39	✓
In-House / Comma.ai	✓	✓			✓
Comma.ai / Udacity	✓	✓	✓	✓	✓
Comma.ai / In-House	✓	✓	✓	0.05	✓

D. Adversarial Attack

In addition to detecting out-of-distribution data, we also seek to detect data that looks more similar to the training distribution but yet produces an erroneous output due to adversarial perturbation. First we explore FGSM attacks by strategically modifying input images, which assumes white-box access. Figure 6 demonstrates that our task-aware model is able to detect when an input becomes more perturbed (i.e., the novelty value, along with the prediction error, increases with the level of perturbation). We compare FGSM attacks on multiple variants of generative models, as shown in Figure

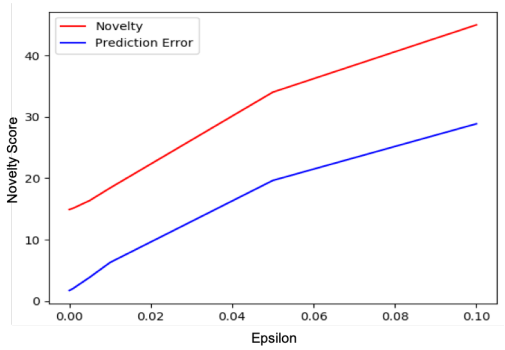

Fig. 6. Novelty plotted with prediction error, demonstrating similar trends. As the ϵ-attack value increases, which means a greater perturbation is being made to the input image, the prediction error increases along with our quantification of novelty.

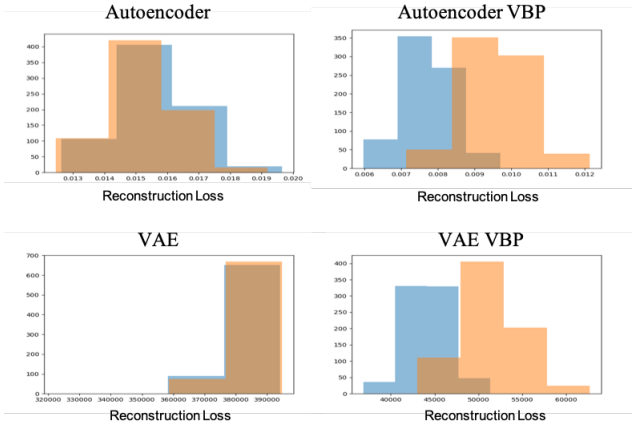

Fig. 7. Adversarial noise attack on the two comparison methods where the orange is the training set and blue is the test set. This figure can be compared with the task-aware result from Figure 4. Since there is significant overlap between the training and test set, we are not able to detect novel inputs. The x-axes differ between each method because each might have different loss metrics and might be trained on different input (normal vs. VBP images).

7. FGSM attacks at low ϵ values are imperceptible to the eye, as shown in Figure 1, and yet these subtle changes can affect the prediction network. We find that autoencoder based methods alone cannot distinguish between attack images and the training data. This is because the model is not task-aware. The addition of VBP helps to separate the training and novel images but there is still significant overlap between the novelty distributions [21].

Finally, we also experiment with a black-box approach to adjustments the input image. In particular, we explore realistic changes to input images which are not necessarily adversarial in nature. These changes include increasing the brightness to varying degrees to increasingly wash the image out like the sun would on a bright day. We demonstrate in Figure 8 through a similar experiment as Figure 6 that our task-aware method can detect changes in brightness, test images are shown in Figure 9. Our findings that brightness can affect the prediction module aligns with findings by [6].

V. Conclusion

In this paper, we presented a task-aware novelty detection framework in which key features and characteristics that are learned in the prediction network leveraged through network

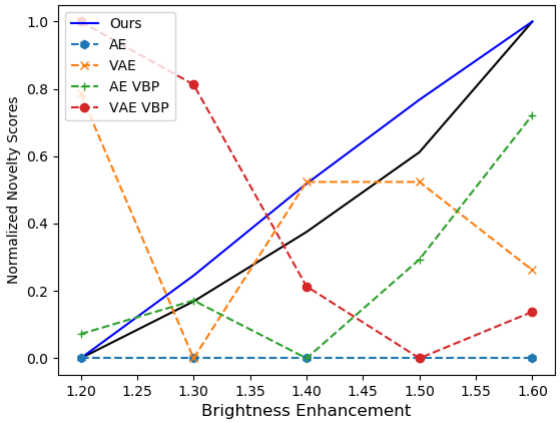

Fig. 8. The effect of brightness changes on normalized predicted $n(x)$ by our task-aware method against normalized reconstruction losses by comparison methods. The solid black line is the *actual* prediction error when compared against the original image, which increases with brightness added. We find that generative methods are not able to recognize the changes in brightness, which can affect output prediction. A potential reason is that mean squared error loss is not particularly sensitive to these changes [24].

Fig. 9. A visualization of the brightness changes on an example Udacity image where a brightness enhancement of 1.3 on the original image is clearly visible to the human eye. And yet, generative models were not able to detect the difference in Figure 8.

saliency maps for novelty detection. We show that the intrinsic tie between learned features and accuracy of the predicted action can facilitate the detection novel situations where unfamiliar images or modifications to the input could cause erroneous predictions. We applied our method on a number of public datasets and an in-house dataset, demonstrating that we are able to match prior work on detecting out-of-distribution data as well as surpassing their ability to detect adversarial attacks which include white and black-box attacks. While our method performs well empirically on a number of real-world datasets, we plan to develop more concrete theory to characterize novelty for real-world deep-learning models. We envision application layers developed on top of our method which would utilize these novelty scores to determine system behavior. In applications like self-driving cars, engineers might set a novelty score tolerance. Once surpassed, the car would revert to more conservative driving behavior to account for the fact that the trained model may not produce a trustworthy prediction. As deep-learning models are adopted into everyday systems, our model contributes towards efforts to improve safety.

Acknowledgments

This research is based on work supported in part by NSF grants 1521523, 1715154, and 1763399. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of sponsors.

REFERENCES

[1] A. J. Hawkins. (2018) Inside waymo's strategy to grow the best brains for self-driving cars. [Online]. Available: https://www.theverge.com/2018/5/9/17307156/google-waymo-driverless-cars-deep-learning-neural-net-interview

[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[3] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *International Conference on Learning Representations Workshop Track*, 2017.

[4] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "A rotation and a translation suffice: Fooling cnns with simple transformations," *International Conference on Machine Learning*, 2019.

[5] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[6] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*. ACM, 2018, pp. 303–314.

[7] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[8] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, p. 18, 2010.

[9] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics Science and Systems XIII*, 2017, pp. 64–72.

[10] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3379–3388.

[11] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018, pp. 4390–4399.

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

[13] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[14] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *International Conference on Learning Representations*, 2017.

[15] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *International Conference on Learning Representations*, 2018.

[16] R. McAllister, G. Kahn, J. Clune, and S. Levine, "Robustness to out-of-distribution inputs via task-aware generative uncertainty," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2083–2089.

[17] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[18] A. T. Nguyen and E. Raff, "Adversarial attacks, regression, and numerical stability regularization," *AAAI Workshop on Engineering Dependable and Secure Machine Learning Systems*, 2019.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Conference on Learning Representations*, 2015.

[20] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. J. Ackel, U. Muller, P. Yeres, and K. Zieba, "Visualbackprop: Efficient visualization of cnns for autonomous driving," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[21] V. Chen, M.-K. Yoon, and Z. Shao, "Novelty detection via network saliency in visual-based deep learning," *2019 IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, pp. 52–57, 2019.

[22] "Udacity self-driving car dataset," https://github.com/udacity/self-driving-car/tree/master/datasets.

[23] "Comma.ai driving dataset," https://github.com/commaai/research.

[24] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel, "Learning to generate images with perceptual similarity metrics," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 4277–4281.