InterpNet: Interpretability for Autonomous Driving Neural Networks

Raghav Kapoor
Santa Clara University
Santa Clara, California, USA
rkapoor@scu.edu

Casey Nguyen
Santa Clara University
Santa Clara, California, USA
cnguyen5@scu.edu

David Anastasiu
Santa Clara University
Santa Clara, California, USA
danastasiu@scu.edu

Figure 1. This figure illustrates a sample frame from the Corl2017 Benchmark Test in the Carla Driving Simulator.

Abstract

Deep Neural Networks have demonstrated impressive performance in complex tasks, such as image classification and speech recognition. However, due to their multi-layer structure combined with non-linear decision boundaries, it is hard to understand what makes them arrive at a particular output. In many state-of-the-art interpretability solutions for vision-based models, a passive method is applied after the model is trained. This means that no modifications are made to the network until after it is fully trained.

Therefore, we propose InterpNet, an active interpretability method for autonomous driving neural networks that operates by adding a penalty function in the convolutional layers of the network. This method is intended to prevent the model from developing complex data representations that are not human-understandable by creating sparse feature maps that can ignore noise. Through the use of an autonomous driving simulator and feature extraction methods, we proved that our regularized model was more effective and interpretable than the baseline version. Specifically, our results show that our regularized model was more autonomous and learned more compressible and less random features than the baseline model.

CCS Concepts: • **Computing methodologies** → **Vision for robotics**; **Interest point and salient region detections**; **Regularization**.

Keywords: neural networks, autonomous driving, interpretability, regularization, sparsity, Keras, Tensorflow, deep learning

ACM Reference Format:
Raghav Kapoor, Casey Nguyen, and David Anastasiu. 2022. InterpNet: Interpretability for Autonomous Driving Neural Networks. In *KDD Undergraduate Consortium, August 14–18, 2022, Washington, DC, USA*. ACM, New York, NY, USA, 7 pages.

KDD-UC, August 14–18, 2022, Washington, DC, USA

1 Introduction

1.1 Motivation

The problem with neural networks in modern-day technology is that they are considered black boxes. When the performance of the model is substandard or unexpected, machine learning engineers have a difficult time figuring out why the neural network produced a specific prediction. This can be especially problematic in the domain of autonomous driving, since the lack of understanding of a neural network's incorrect decision can present a significant negative impact on the safety of the passengers in those vehicles.

For example, in 2018, a Tesla Model X crashed into a cement wall on the freeway [10], and the company could not understand the reasoning behind the car's movements until long after the accident had occurred. With an interpretability method, a company could trace the path that the data takes in providing a driving decision. This tracing could take place before or immediately after the actual deployment of the self-driving system. This would help reduce the probability of additional such accidents happening in the real world.

1.2 Challenges of Implementing Neural Network Interpretability in Self-Driving Vehicles

There are two main reasons why autonomous systems are not inherently interpretable as indicated by Zablocki et al. [12]:

1. Deep learning models may potentially face the limitations due to datasets that contain numerous biases, are too general, and are not properly curated. This leads the system to learn from spurious correlation and overfit to certain situations.
2. Self-driving systems have to solve incredibly complex problems. For humans, it may be simple, but it is very difficult for a system to solve related tasks with different environments. Thus, the model that led to the prediction is very chaotic and difficult to navigate for humans.

Autonomous driving is a high-stake, safety-critical application. From a societal point of view, performance guarantees should be mandatory. However, there are many scenarios where self-driving models are not testable due to the impossibility of listing and evaluating every scenario that a model can encounter. A solution to this issue is to be able to explain a model's decision making process in making certain decisions in a given scenario and pinpoint the area of error.

1.3 Solution

To this end, we propose a solution to improve the interpretability of neural networks on autonomous cars. Our method is applied as the model is learning, such that the learning process does not compromise prediction accuracy. To verify this, a feature visualization method is executed to determine what features/concepts the model has learned based on how input data is transformed by the neural network.

2 Related Work

Within the domain of Explainable Artificial Intelligence, various neural network interpretability methods have been defined and are organized using the taxonomy defined by Zhang et al. [13]. Within this taxonomy, interpretability techniques can be decomposed into three dimensions.

The first of these three dimensions specifies the type of engagements. This is further broken down into passive and active. Passive engagement introduces interpretability after model training. Active engagement introduces interpretability behavior during model training and actively influences the network or training process

The second dimension is type of explanation. This is decomposed into the following: examples, attribution, hidden semantics, or rules.

The third dimension is the focus of the interpretability method. This is categorized as local, semi-local, or global. Local focus seeks to explain network's prediction based on an individual sample. Semi-local focus attempts to explain a group of similar inputs. Lastly, global focus strives to explain the network as a whole.

2.1 Passive Interpretability

Woh and Liang [7] used a technique from statistics known as influence functions, which are capable of indicating how a change in one data instance can impact an overall estimator. Within this particular work, Woh and Liang applied influence functions to trace the path taken by a training data instance from the model's prediction through the model and back to the original training instance. In the empirical evaluation of this approach, Woh and Liang demonstrated how they were able to understand model behavior, assess model vulnerability to adversarial training examples, debug models, and detect dataset errors. However, since this model mainly focuses on the impact of individual training points towards a model's prediction output under the assumption of minimal model change, it does not possess the capability of explaining global changes made to a network.

Bojarski et al. [2] produced a new method in the field of neural network interpretability of autonomous driving called VisualBackProp. VisualBackProp visualizes and determines which set of pixels of the input image made the most contributions to the prediction made by the convolutional neural network. Bojarski et al's method was shown to be computationally competent as well as able to provide efficient and accurate visualizations. Although this method is very fast and applicable to real-time application, it does not provide any practical information for debugging the original autonomous driving neural network.

2.2 Active Interpretability

Wu et al. [11] created a new tree regularization method to make deep models that can be closely modeled by decision trees with a few nodes, so that the deep models can be human-simulatable. This tree regularization method takes the form of a true-average-path-length penalty function, such that the deep model would still maintain its accuracy and avoid complexity, which is defined as long average path length. The authors discovered that tree regularized models were able to outperform conventional regularization methods such as L1 norm and L2 norm in accuracy, while maintaining interpretability through the use of decision tree proxies. Even with these exemplary results though, this tree regularization technique is limited only to input features that are interpretable and would not work on other types of input features, such as pixel data from images.

Alvarez-Melis and Jaakkola [1] created a bottom-up interpretable model that maintains desirable characteristics of simple linear models in terms of features and coefficients without limiting performance. They made a self-explaining neural network model (SENN) that progressively generalizes linear classifiers to complex architecturally explicit models. Instead of looking at single pixels on predictions, their model aims to examine the higher-level features because individual pixels tend to be hard to analyze and often lead to chaotic explanations. The results showed that they were able to create complex models with robust explanations.

2.3 Hybrid Interpretability

Plumb et al. [8] designed a hybrid approach known as EXPO. Within this method, a domain-independent regularization technique is applied to black-box models, which provides greater control over the quality of post-hoc explanations generated to elucidate the logic used by the model. The results of the application of EXPO regularization to increase fidelity and stability show that this method slightly improves model accuracy and greatly improves the quality of interpretability across each of the datasets. However, this method does not provide a resolution to the issue of vulnerability of local explanations to adversarial training examples and does not provide results when trained on data without semantic features.

Dong et al. [4] built a novel neural network architecture specifically within the domain of autonomous driving known as a global soft attention model. The results produced by the authors support the accuracy improvement brought by the global soft attention model compared to the baselines due to its focus on global features, the fusion of individual pieces of information, and the ability to capture long-range correlations in the input data. However, even with the introduction of some form of a set of explanations for the logic used by this architecture, the authors did not evaluate the quality of these explanations.

3 Method

3.1 Dataset

To train the end-to-end self-driving model, the Udacity Self Driving Car Dataset 3-1: El Camino [6] is used. This dataset consists of 3 hours of driving data from the Udacity office in Mountain View to San Francisco and from San Francisco back to the Mountain View Udacity office.

This dataset is then preprocessed to only contain the center camera view with the corresponding steering wheel angle. Furthermore, the dataset images in which the car is stationary either at the beginning or the end of the dataset are removed, resulting in a final dataset size of 210,000 data points. The first 190,000 of these points are used for training the model and the last 20,000 are used as a validation set, to verify that the model is not overfitting to/memorizing the training data labels. Further preprocessing is applied to the center camera images, by resizing the input images from 480x640 (height x width) to 125x349, converting the dataset image representation from an 8-bit unsigned integer to Tensorflow's 32-bit floating-pointing number, and converting the images from RGB to grayscale. These images are then fed into the self-driving model.

3.2 Self-Driving Model

The end-to-end driving model used in this work was developed by Bojarski et al. [2] called NetHVF. This model was initially designed as a means of validating the VisualBackProp approach of extracting feature maps that accurately represent the concepts learned by a vision-based neural network. However, in this project, the application of this model is extended beyond its original scope and evaluated for its driving capabilities. The composition of this model is provided in Table 1.

Table 1. Architecture of NetHVF

	NetHVF		
Layers	Layer Output Size	Filter Size	Stride Size
conv	32 x 123 x 349	3 x 3	1 x 1
conv	32 x 61 x 173	3 x 3	2 x 2
conv	48 x 59 x 171	3 x 3	1 x 1
conv	48 x 29 x 85	3 x 3	2 x 2
conv	64 x 27 x 83	3 x 3	1 x 1
conv	64 x 13 x 41	3 x 3	2 x 2
conv	96 x 11 x 39	3 x 3	1 x 1
conv	96 x 5 x 19	3 x 3	2 x 2
conv	128 x 3 x 17	3 x 3	1 x 1
conv	128 x 1 x 8	3 x 3	2 x 2
FC	1024	-	-
FC	512	-	-
FC	1	-	-

Note the following details about the design of the NetHVF model. Each layer except for the last feedforward (FC) layer in Table 1 is followed by ReLU activation. Each convolutional (conv) layer is preceded by a batch normalization layer. Let n be the number of feature maps, h be the height and w be the width. For conv layers, layer output size is n × h × w. Filter size and stride are given as h × w.

3.2.1 Regularization. To make this model more interpretable, our proposed method involves applying active model interpretability in the form of the penalty function on the model's optimization function, also known as regularization. Specifically, L1 Norm Regularization was applied mainly to the convolutional layers of the NetHVF model. The purpose of this is to induce sparsity in the features learned by the network, such that the reduction in parameters would allow the model to be more selective in the features that it learns during training. Through this process of selecting the features learned in each of the convolutional layers, the model may be able to filter out some unnecessary noise that may be irrelevant to autonomous driving and learn identifiable patterns. This in turn may make the features learned by the model more interpretable for humans.

However, from a theoretical point of view, the application of L0 Norm Regularization would be more ideal, since it can guarantee sparsity. As shown by the graphical depictions of L0 and L1 Norm Regularization in Figure 2, L0 Norm Regularization is axis-aligned, which means that this regularization will constrain optimal solutions of the model to be axis-aligned. However, since solving a constrained optimization problem with L0 Norm Regularization is NP-hard, L1 Norm Regularization is used instead where optimal sparse solutions are located at the vertices of its graphical representation.

(a) (b)

Figure 2. Figures (a) and (b) represent the graphical representation of L1 Norm Regularization and L0 Norm Regularization respectively.

3.3 Test Environment

For model evaluation, we decided to use the open-source driving simulator CARLA [5]. This simulator has a diverse set of driving environments with a flexible API for greater ease in adding different vehicle sensors. In addition, CARLA has a comprehensive driving benchmark, Corl2017, to evaluate autonomous driving models in two towns, Town01 and Town02, with 24 different experiments run in both towns. Within these experiments, autonomous driving models are evaluated for being able to drive straight, make a single turn, go to an arbitrary position, and go to an arbitrary position with random moving objects in the environment. For each of these tasks, the weather conditions of the simulation environment are changed to the following: clear noon, heavy rain noon, clear sunset, after rain noon, cloudy after rain, and soft rain sunset. In this work, only the Town02 simulations are run, since Town01 is used in the training dataset that CARLA provides and not utilized to train the baseline and regularized NetHVF models.

3.4 Data Collection

During the benchmark test, the number of human interventions, autonomous driving time, and images viewed by the models are collected. For the number of human interventions, this quantity is incremented based on observations made when the simulated car with the loaded model drifted out of a lane or crashed into a building/pole during each of the simulation episodes. The autonomous driving time is measured by taking the total time during which the simulated car with the loaded model drives continuously until the car is no longer able to navigate after crashing or for the duration of the simulation. Lastly, the images viewed by the models, these images are collected every 50 frames.

4 Experiment Results

4.1 Trained Models

In search of the optimal baseline and the L1-Regularized models, various model configurations were trained to optimize the Mean Squared Error Cost Function ($J(\theta)$) between the predicted steering wheel angle ($h_\theta(x^{(i)})$) and ground-truth steering wheel angle ($y^{(i)}$), as shown below.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \tag{1}$$

A total of 10 model configurations were evaluated for the baseline model and 30 model configurations for the L1-Regularized model (InterpNet). After training these models until convergence, the optimal solution for each configuration was determined based on the epoch at which minimal validation loss was achieved. For the baseline, the optimal model was that trained for 52 epochs with a batch size of 32 and a learning rate of 5e-4. This model achieved a validation loss of 0.00255298. For InterpNet, the optimal model was that trained for 260 epochs with a batch size of 32, a learning rate of 1e-4, and an L1 Norm regularization strength of 1e-3. This model achieved a validation loss of 0.00200723. The training plots for optimal baseline and InterpNet model are shown in Figure 3.

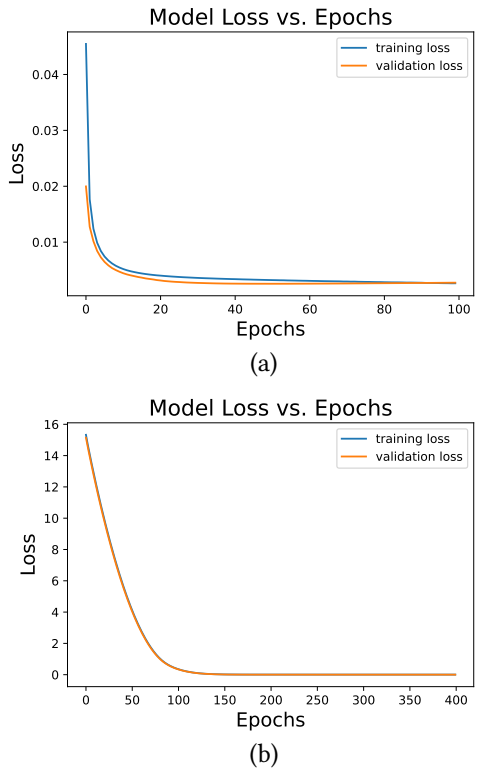

Figure 3. Figures (a) and (b) depict the training plots of the optimal baseline and InterpNet models respectively.

4.2 Autonomy

To measure model effectiveness, the autonomy of the models was measured using the number of human interventions that were recorded during the 10-hour benchmark test. The precise formulation of this metric, as stated by Bojarski et al. [3], is provided below.

$$autonomy = 1 - \left(\frac{number\ of\ interventions \times 6\ seconds}{elapsed\ time\ [seconds]} \right) \tag{2}$$

For further clarification, the autonomy of the model was calculated indirectly by measuring how non-autonomous the model is. This non-autonomy portion of the metric was calculated by taking the number of human interventions multiplied by 6 seconds over the total elapsed driving time. The number of human interventions was multiplied by 6 seconds because it takes 6 seconds on average for a person to correct a driving error.

Based on the data collected on the number of human interventions from the Corl2017 benchmark tests for the baseline model and InterpNet, the autonomy results are shown in Table 2.

As shown in Table 2, the baseline model drove autonomously for a smaller amount of time compared to the InterpNet model and required more human interventions as well. In total, this equated to the InterpNet model being more autonomous than the baseline model by 3%.

Table 2. CARLA Autonomy Measurement

	baseline	InterpNet
Number of Interventions	618	598
Autonomous Driving Time (seconds)	2870	3090
Autonomy	43%	46%

4.3 Interpretability

To measure model interpretability, the compressibility and randomness of the features learned by both models. As indicated by Samek et al. [9], feature maps that are more compressible are more likely to align with an observable pattern, which indicates interpretability. Feature map randomness is measured as a second means of verifying model interpretability, since less random feature maps are more likely to align with a particular pattern. To extract these feature maps, the VisualBackProp approach [2] was used with the images collected during the benchmark test. Histograms were then created to describe feature compressibility and feature randomness of the models using Principal Component Analysis (PCA) and entropy respectively.

4.3.1 Feature Compressibility. The histograms for measuring the compressibility of the features learned by the baseline model and InterpNet are shown below in Figure 4.

From these histograms, it is observed that the distribution of the compressibility of the features learned by the baseline model closely resembles a Gaussian distribution with a greater amount of variation compared to the distribution of the compressibility of the features learned by InterpNet. Additionally, the peak compressibility of the learned features of the baseline model requires 7 Principal Components compared to the consistent 3 Principal Components for learned features of InterpNet. This indicates that the features learned by InterpNet are generally more compressible than those learned by the baseline model.

4.3.2 Feature Randomness. The histograms for measuring the randomness of the features learned by the baseline model and InterpNet are shown below in Figure 5.

From these histograms, a similar trend is observed as from the histograms of the features learned by both models. Specifically, the peak randomness of the learned features of the baseline model falls within the range of 0.9 and 1.0 compared to the 0.5 and 0.6 range for learned features of InterpNet. This indicates that the features learned by InterpNet are generally less random than those learned by the baseline model. However, the distribution for the randomness of the features learned by the baseline model is skewed to the right/maximum entropy instead of closely resembling a Gaussian distribution. Nonetheless, this still indicates that

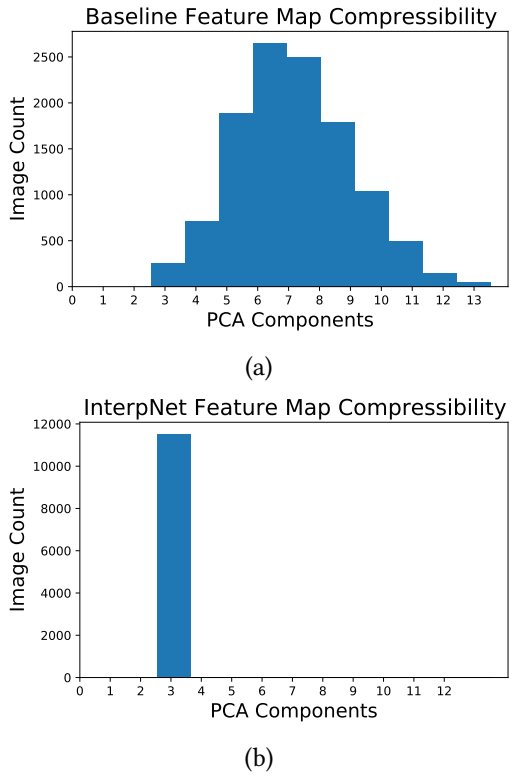

Figure 4. Figures (a) and (b) depict the number of PCA components required to compress the feature maps extracted from the optimal baseline and InterpNet models respectively. The PCA algorithm was configured to compress the learned features while still capturing 90% of the variance within the original features.

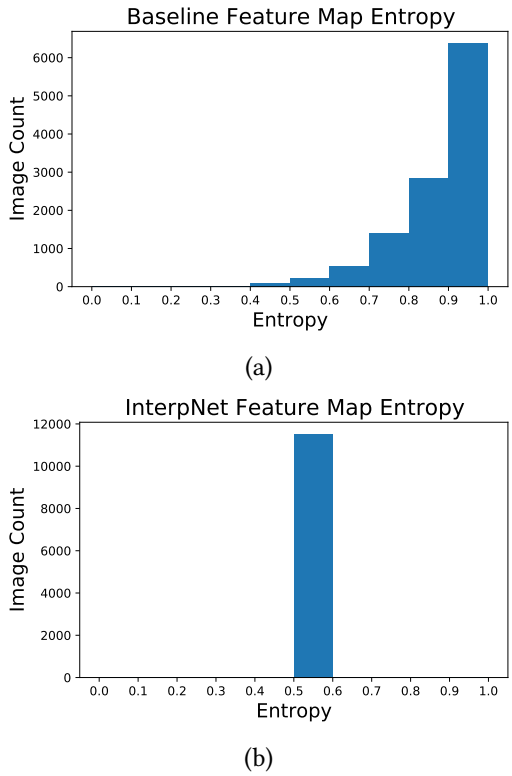

Figure 5. Figures (a) and (b) depict the entropy of the feature maps extracted from the optimal baseline and InterpNet models respectively. These entropy values range between 0 and 1 inclusive, in which 0 indicates no randomness and 1 indicates complete randomness.

the features learned by InterpNet are generally less random than those learned by the baseline model.

5 Ethical Concerns

As software continues to play a key role in modern-day vehicles, machine learning engineers working in this domain must ensure that their solutions address prevalent ethical concerns. Even with the optimistic outlook on self-driving cars, many bring up unsolvable decision making problems like the trolley problem. The solution to the trolley problem is extremely limited, and all answers can be perceived as ethically wrong. A typical approach to solving this problem is found by analyzing it through various ethical theories, such as utilitarianism. This theory stresses the greater good meaning that the ethically correct answer would be to choose the least amount of casualties in order to maximize utility. Depending on the ethical framework, different theories can be used to justify a decision. However, these conclusions are only drawn by humans. How do these systems come to their conclusion?

6 Conclusion and Future Work

In this work, we have introduced InterpNet, an active interpretability method to encourage an autonomous driving neural network to learn more interpretable features. Through the use of an autonomous driving simulator, we have demonstrated that our approach not only is more autonomous but also more interpretable based on the general decrease in required principal components and entropy of the extracted feature maps compared to our baseline model. While we have demonstrated improvements in model effectiveness and interpretability through our method, future work could continue towards examining the impact of additional regularization terms. One such improvement would include the use of regularizing for mutual information along with the L1 Norm in the convolutional layers. This would constrain the model's optimal parameters to be sparse and independent between neurons and across network layers. Furthermore, regularization can be applied to the feedforward layers to more easily trace through the model's decision-making process. This could be done using tree regularization to produce an accurate decision tree proxy, as described by Wu et. al. [11].

References

[1] David Alvarez Melis and Tommi Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[2] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry J Ackel, Urs Muller, Phil Yeres, and Karol Zieba. 2018. VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving. (2018), 4701–4708.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. 2016. End to End Learning for Self-Driving Cars. https://arxiv.org/abs/1604.07316

[4] Jiqian Dong, Sikai Chen, Shuya Zong, Tiantian Chen, and Samuel Labi. 2021. Image transformer for explainable autonomous driving system. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2732–2737.

[5] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*. 1–16.

[6] MacCallister Higgins. 2016. Udacity Self Driving Car Dataset 3-1: El Camino. https://academictorrents.com/download/c9dae89d2e3897e6aa98c0c8196348c444998a2a.torrent

[7] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1885–1894.

[8] Gregory Plumb, Maruan Al-Shedivat, Ángel Alexander Cabrera, Adam Perer, Eric Xing, and Ameet Talwalkar. 2020. Regularizing Black-box Models for Improved Interpretability. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 10526–10536. https://proceedings.neurips.cc/paper/2020/file/770f8e448d07586afbf77bb59f698587-Paper.pdf

[9] Wojciech Samek, Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems* 28 (11 2017), 2660–2673.

[10] Ars Technica Timothy B. Lee. 2020. New data backs up details in a fatal 2018 Tesla Model X Crash.

[11] Mike Wu, Michael Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. 2018. Beyond Sparsity: Tree Regularization of Deep Models for Interpretability. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://ojs.aaai.org/index.php/AAAI/article/view/11501

[12] Éloi Zablocki, Hedi Ben-younes, Patrick Pérez, and Matthieu Cord. 2021. Explainability of vision-based autonomous driving systems: Review and challenges. *ArXiv* abs/2101.05307 (2021).

[13] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. 2021. A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 5 (2021), 726–742. https://doi.org/10.1109/TETCI.2021.3100641