Article

Improving Performance of the PRYSTINE Traffic Sign Classification by Using Perturbation-based Explainability Approach

Kaspars Sudars, Ivars Namatēvs*, and Kaspars Ozols

Institute of Electronics and Computer Science, Dzerbenes str.14, Riga, LV-1006, Latvia;
* Correspondence: ivars.namatevs@edi.lv; Tel.: +371 26133567

Abstract: Model understanding is critical in many domains, particularly those involved in high-stakes decisions, i.e., medicine, criminal justice, and autonomous driving. Explainable AI (XAI) methods are essential for working with black-box models such as Convolutional Neural Networks. This paper evaluates the traffic sign classifier of Deep Neural Network (DNN) from the Programmable Systems for Intelligence in Automobiles (PRYSTINE) project for explainability. The results of explanations were further used for the CNN PRYSTINE classifier vague kernels` compression. After all, the precision of the classifier was evaluated in different pruning scenarios. The proposed classifier performance methodology was realised by creating the original traffic sign and traffic light classification and explanation code. First, the status of the kernels of the network was evaluated for explainability. For this task, the post-hoc, local, meaningful perturbation-based forward explainable method was integrated into the model to evaluate each kernel status of the network. This method enabled distinguishing high and low-impact kernels in the CNN. Second, the vague kernels of the classifier of the last layer before the fully connected layer were excluded by withdrawing them from the network. Third, the network's precision was evaluated in different kernel compression levels. It is shown that by using the XAI approach for network kernel compression, the pruning of 5% of kernels leads only to a 1% loss in traffic sign and traffic light classification precision. The proposed methodology is crucial where execution time and processing capacity prevail.

Keywords: Explainable AI, Convolutional Neural Network, Network Compression;

1. Introduction

The nature of road traffic is currently considered a complex and dynamic environment, where safety performance depends on several interconnected factors. Failure to comply with them would be the cause or interaction to cause the road accidents [1]. The primary reasons for the accidence are drivers' characteristics, the vehicle itself, and road characteristics [2]. One of the modern concepts related to road infrastructure is "self-explaining roads" [3]. The concept aims to convey information about the upcoming situation to the driver or the visual system of a self-driving car in a comprehensible, faithful, and trustful way, using various measures related to traffic signs and road markings. Taking these specifics for granted is an intrinsic part of road control infrastructure and provides drivers with the necessary information, warning of road regulations, and keeping pedestrians safe.

For autonomous driving, accurate and robust perception of traffic signs is essential for motion planning and distinctness. However, traffic sign detection and classification problem are still challenging due to the following reasons: (1) traffic signs are easily confused with other objects in streets scenes, (2) weather conditions, time of the day, refection, and occlusions would decrease the classification performance, (3) the size, shape, and colour of the traffic signs, (4) slight inter-class variance due to similar appearance of signs [4].

Therefore, traffic sign detection and classification has been just now studied extensively in the computer vision community. More recently, the availability of annotated

large datasets [5,6] and computational gains with powerful GPU cards [7] show good results by using convolution neural networks (CNN)s [8, 9]. For this purpose, to classify all specific traffic signs and traffic light classes, a classifier based on a CNN architecture from the Horizon 2020 ECSEL-JU project "Programmable Systems for Intelligence in Automobiles" (CNN PRYSTINE) was for providing our experiment considered [10]. Despite the success achieved by this approach, the existing CNN classifier is still lacking because it provides a classification output that does not provide us with what information in the input data makes them arrive at their decisions. It is known that CNNs consist of a highly complex internal structure and, as a result, is very difficult to explain because of their black-box nature. It leads to a challenge to understand what exactly goes on at each layer. It is also known that after training, each layer progressively extracts higher and higher-level features of the image until the final layer essentially predicts what the image shows. We have focused on using explainable AI (XAI) to identify novel higher-level patterns and detections to provide more precise classification strategies to overcome these limitations. Moreover, the explanations will ideally let us comprehend the model's reasoning behaviour and understand why the model has predicted explicit decisions, e.g. to classify the traffic sign in a specific manner or associate certain properties with the performance of CNN.

Despite the advantages, the CNN models require a significant number of resources including, processing capacity, energy, bandwidth, and storage capacity. Therefore, different CNN compression techniques have been proposed in the existing literature to mitigate these shortcomings, i.e., network pruning, sparse representation, bits precision, knowledge distillation, and miscellaneous [11]. The key reason to tackle this problem is to find an appropriate method for CNN compression by using the XAI approach.

This paper presents a study on how explainability at a model level can be utilised on network pruning and how pruning influences prediction precision. We show that the perturbation-based methods can be used not only to explain the decision of CNN but can be used as a perspective tool for compressing CNNs with minimal precision compromise.

The following section discusses the perturbation-based explanation methods appropriate for network parameter evaluation. Section 3 presents the overview of the used CNN PRYSTINE network architecture for the classification of traffic sign and traffic light. Section 4 describes the mathematical background for the proposed methodology. Section 5 shows the results of the experiment. Finally, Section 6 presents the discussions and future directions as well as conclusions.

2 Perturbation-based methods

For black-box models like CNNs, it is crucial to make the prediction process faithful to a particular model by explaining why it reaches such results. Current studies into algorithmic explanation methods for predictive models fall into three main approaches: attribution, distillation, and intrinsic. The attribution focuses on measuring the attribution or feature relevance scores. The distillation focuses on reducing the complexity of CNN models by transforming them into simple, easily understandable surrogate models. Finally, the intrinsic approach integrates the inner states of the deep networks or modular algorithms to justify the model. First, the attribution approach can be divided into three subcategories: perturbation-based, functional, and structural explanations. The first subgroup's methods are analysed here, and the eligible has been chosen to support our experimental methodology.

The idea of perturbation-based explanations is to calculate the feature importance of a feature or a group of features in a specific model by simulating a lack of knowledge about the value of the feature(s) [12]. In other words, the perturbation methods try to assess attribution or feature relevance by testing the model's response to the removal, masking or altering the feature and measuring the corresponding feature relevance scores. Perturbation-based explanations are widely used with arbitrary prediction models and support explanations of individual predictions. These methods express an explanation by

manipulating the input image and/or activations of a CNN [13]. They can also visualise the predictive model as a whole. When a perturbation highlights image regions, it has an easily explainable meaning, i.e., editing that region in the actual input will significantly affect the model's prediction. They have the advantage of a straightforward explanation, as they directly measure the marginal effect of some input features on output [14]. In this case, perturbation-based methods only require the propagation of one forward and/or backward pass through the CNN to generate an attribution visualisation. Understanding the visual perception aspects captured within a deep model has become particularly relevant in the context of an explanation of deep networks. Our experiment applied saliency to highlight essential regions in the input image to understand CNN inference.

The Occlusion Sensitivity method of Zeiler and Fergus [15] is based on dividing the input into segments called patches, masking them, and measuring the input impact of each defined patch on the classification score. For example, an image can be split into a grid of regular non-overlapping patches, and a mask slid over an image covering patches. The authors occluded different segments of an input image with a grey patch and visualised the change in the activations of the later layers. When the patch covers the critical area, the output prediction performance drops significantly. A similar approach was proposed by Zhou et al. [16], using small grey squares to occlude image patches (in a dense grid) to explain scene classification. The visualisation depicts the sensitivity area of an image for its classification label. The Meaningful Perturbation method proposed by Fong and Vedaldi [17] uses the output value of the DNN changes as the input is penalised by deleting specific regions. Attribution aims to identify which regions of an image are used to produce the output value. The idea is not to iterate over all possible perturbations but to search locally for the best perturbation mask, i.e., the smallest deletion mask. The authors have considered three perturbation types for creating a perturbation mask: replacing the input region with a constant value, injecting noise, or blurring the image. Extremal perturbations are regions of an input image that maximally affect the activation of a particular neuron in a DNN. The Extremal Perturbations method [18] optimises the perturbations by choosing smooth perturbations masks, maximising the classifier's output confidence score. Randomised Input Sampling for Explanation (RISE) [19] explains DNN black-box models by estimating pixel saliency importance (importance map) of the input image regions. The importance of pixels is estimated by blurring them in random combinations, reducing their intensities to zero, and weighting their changes in the output by occluding patterns. The authors [20] developed a fast saliency detection method Universal Adversarial Perturbations for image classifiers by manipulating the scores of classifiers by masking salient parts of the input image.

From those mentioned above, the perturbation-based methods try to evaluate the importance of input segments, regions or pixels on the classifier's decisions, and how the deep network reacts to changes in the input. For example, if a specific part of the input is masked, how does it affect a classification prediction. Therefore, the relevance (attribution) score evaluates the strength of the connection between the pixel or group of pixels to the specific network output. From this point, we can hypothesise that certain regions on an image are not involved in classifiers decision-making. Consequently, it leads to an assumption that there might be parameters in the network that if they are excluded from the deep network, the classifier's prediction precision remains at the same level.

In our experiment, we used the meaningful perturbation method to blackout the region of traffic sign in the feature map of the last 4-th layer before a fully connected layer of CNN PRYSTINE classifier model to distinguish high and low-impact parts of CNN. Using the perturbation, we would have explained the attribution of the 4-th layer kernels on classifier prediction. Therefore, this approach allowed us to identify those vague kernels that were not involved in traffic sign and light classification and compressed them.

3 Implementation of Traffic Sign and Traffic Light Classifier

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2021 doi:10.20944/preprints202111.0186.v1

The architecture of a 5-layer CNN from the PRYSTINE project [21] has been used for traffic sign and light classification. Each layer of the network consists of convolution filtering, batch norming, sigmoid activation, and downsampling by max pooling. The last, 5-layer, consists of a linear classifier with a soft-max, producing the network's output of 45 classes. The precise definition and description with a code of CNN can be found online at GitLab [22].

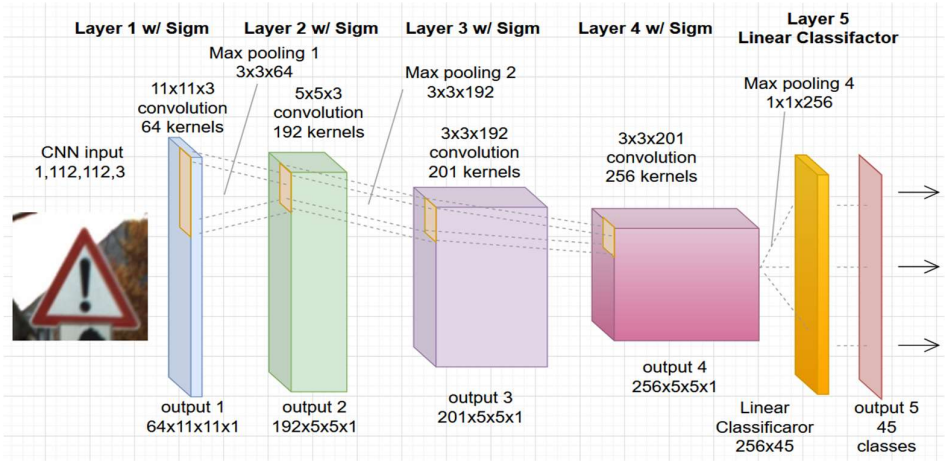

Figure 1. The architecture of the used 5-layer CNN road sign and light classification from the PRYSTINE project.

The network was trained and tested using a multi-class, single-image German Traffic Sign Recognition Benchmark (GTSRB) database [23]. The GTSRB includes more than 50 000 single traffic sign images from 43 classes. The traffic signs are captured at different sizes, rotations, and different light levels, thus giving a natural road traffic sign distribution. Next, the two classes of 2239 images from the LISA Traffic Light Dataset (LISA) [24] were added to the GTSRB dataset (adding red/yellow light and green light classes). Finally, the CNN model based on both GTSRB and LISA primary datasets was retrained.

The classification using CNN from the PRYSTINE project provides the precision of 91.86% for traffic sign classification (tested only on GTSRB test data) and the precision of 96.55% for traffic light classification (tested only on LISA test set). The execution time of an image in both classification categories is ~0.0187 sec on Intel Xeon CPU. For explainability testing, the subset of 100 images was randomly taken from the GTSRB and LISA test sets (test.py) and is available together with the developed code to approve our experimental methodology.

4. CNN Layer Perturbation-based Forward

The standard perturbation approach aims to identify which regions of an input image x_0 are used by the black box CNN $f(x)$ to produce the output value $f(x_0)$. Derived from the meaningful perturbation method, given an input traffic sign image x_0 our goal was to define an explanation for traffic sign and traffic light classification in the output of the last convolution layer (4-th layer) before max-pooling by masking the output region R of the feature map with a constant black value, see Figure 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2021 doi:10.20944/preprints202111.0186.v1

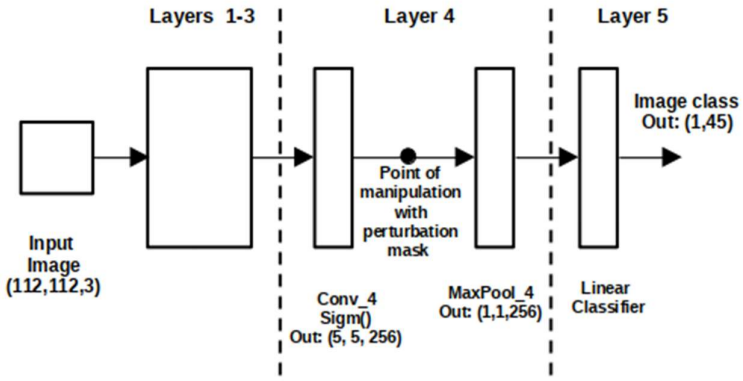

Figure 2. CNN PRYSTINE manipulations with perturbation mask in 4-layer.

Formally, let $m : \Lambda \rightarrow [0.1]$ be a perturbation mask, associating each pixel $u \in \Lambda$ with a scalar value $m(u)$. Then the perturbation operator for a constant value is defined as [17]:

$$[\Phi(x_0; m)](u) = m(u)x_0(u) + (1 - m(u))\mu_0, \quad (1)$$

where μ_0 is an average colour. We use $\mu_0 = 0$, which yields, after normalisation, masking region R by substituting with black colour. Then output vector of the network's 4-th pooling layer $\boldsymbol{p_4}$ are calculated as follows:

$$\boldsymbol{p_4} = pool_4(conv_4(p_3)), \quad (2)$$

where $conv_4()$ is an output of the convolutional 4th-layer from the previous layers. The output vector of the network's 4-th pooling layer, including or excluding particular regions of an image from CNN decision making, is calculated as follows:

$$\boldsymbol{p_4'} = pool_4(conv_4(p_3) * m)), \quad (3)$$

where m is a perturbation mask. Then a coefficient $c_r(n)$ showing n-th feature involvement on CNN decision making of r-th image (where r is an image number and in our experiment $r = 100$ of the test.py set), are calculated as follows.

$$c_r(n) = \begin{cases} 1, if\ p_4(n) = p_4(n)' \\ 0, \qquad if\ other \end{cases} \quad (4)$$

Based on the feature involvement coefficient, the original CNN PRYSTINE model was compared to the model where dedicated parts of the output of the final 4-th convolutional layer of the network were excluded from the CNN traffic sign and traffic light prediction. In our experiment, the region of interest or mask R was a 3x3 region on a 5x5 feature map. This approach further led to explaining the whole prediction power of the classification model. In our experiment, the perturbation mask filters 4-th layer kernels on their impact level accordingly to their significance. After obtaining feature significance statistical involvement in neural network decision-making, kernels of the low impact of the 4-th layer were removed from CNN PRYSTINE, leading to model compression. Therefore, the model was compressed by pruning vague kernels in keeping with its precision calculations. The results of the experiments are presented in the next section.

5 Experiments & Results

This section presents details about the implementation of our experimental methodology. The code for all experiments, including 100 test data set (test.py), is available online at GitLab. The code was developed by using *Python* programming language and *TensorFlow 1.12* machine learning package. Examples of input traffic sign images from the test.py dataset see Figure 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2021 doi:10.20944/preprints202111.0186.v1

(**a**) (**b**) (**c**) (**d**)

Figure 3. Examples of an input image from the test dataset, where (**a**), (**b**), (**c**), traffic sign without occlusion, d occluded traffic sign.

We included in the test.py dataset beside different traffic signs and traffic lights also occluded traffic signs and traffic light images data to check the prediction quality of the CNN PRYSTINE traffic sign classifier. The first three images in a row, (**a**), (**b**), and (**c**), are real traffic signs and are classified correctly. In the false case, (**d**) image, the traffic sign was occluded (covered); as a result, the network did not predict it as a traffic sign. Thus, after the first step, how well our CNN PRYSTINE model classifies, we recognised that our network could provide correct predictions. Next, we have wished to check out explainability by using the meaningful perturbation method of the prediction power of the classifier and how it correlates if CNN PRYSTINE traffic sign and traffic light classifier kernels will be pruned.

The region of a 3x3 perturbation mask was applied onto the 4-th layer feature map before max-pooling of 1x1x1x256 array. Figure 4 depicts an example of the CNN PRYSTINE 4-layer output of 5x5 feature map, where (**a**) a feature map without a mask, but (**b**) a feature mask, where a mask replaces its centre of the 5x5 output with a constant pixel value of zeros (black colour).

(**a**) (**b**)

Figure 4. The output of the 4-th layer feature map of CNN PRYSTINE before max pooling, (**a**) without masking, (**b**) with masking of the central region of 3x3 around the centroid.

To summarise the effect of masking, to know the waveform of the signals to explain the CNN PRYSTINE classifier predictions (correct or wrong), and to know the waveform of the signals that flow through the network model (the signals from the max-pooling layer of 1x1 of the 4-th layer output) the signal characteristics plot are collected, see Figure 5. It shows the characteristics of the signals in a case when the traffic sign is occluded. *DNN pool out* shows the 4-th layer sigmoid activation function values for each of 256 features. It means how much each feature are involved in an input image. Signals *LC weights* and *GT weights* characterise and compare two randomly chosen classes of a linear classifier softmax activation function values. Both signals characterise the linear classifier classification signals, which most correlate with CNN PRYSTINE features.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2021 doi:10.20944/preprints202111.0186.v1

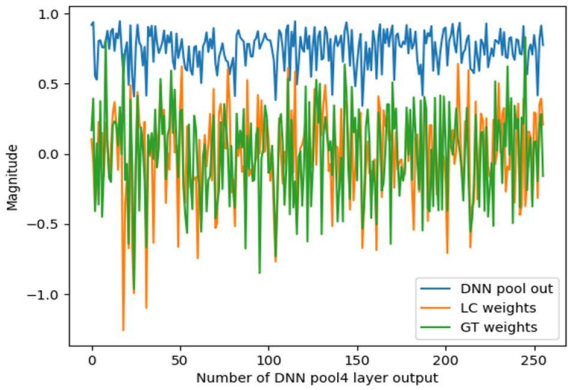

Figure 5. Example of showing output signals of *DNN pool out* of 4-th layer in an occlusion case, linear classifier *LC weights* of chosen class (the most correlated signal), and *GT weights* of the right decision.

The experiment of explainability has shown that the low impact kernels are in 4-layer. When removing them, the processing and storage capacity improves the performance of a CNN PRYSTINE model. Results of the quality of compression are shown in **Table 1**, where the CNN compression rate $\alpha(M, M^*)$ of M^* over M was calculated as follows:

$$\alpha(M, M^*) = \frac{a}{a^*}, \tag{5}$$

where a is the number of kernels in the original model (4-th layer, 256 kernels) and is that of the compressed model a^*. For instance, if eight vague kernels are prune of 256, the compression rate is 3.125%.

Table 1. Experimental results of compressing the CNN PRYSTINE 4-th layer.

Compression rate, %	Precision, %
Original CNN PRYSTINE network	92.0 (total samples 100, correct 92)
3.125 %	92.0
5.47 %	90.0
17.58 %	86.0

As shown from Table 1, using the meaningful perturbation method, if the low-impact kernels are found and removed from the 4-th layer by 3.125% (8 kernels), the network performance improves by the same percentage in the same time the classification precision remains the same. If the compression rate is 5.47% (14 kernels), we still get a competitive advantage on network performance, however, with a slight decline of 2% in classification precision. But if the performance of the 4-th layer is improved by 17.58% (45 kernels are pruned), the loss in precision rate is only 6% compared to the original CNN PRYSTINE model. The proposed methodology is crucial where execution time and processing capacity prevail.

7. Discussion

Every model fails every and then, and when it does, we want explanations *why it* does it. In the case of traffic sign and traffic light classification, when human lives are at stake, we face a problem investigating CNN's inner workings. This study proposed a new methodology based on the XAI approach, which would "whiten" the black-box network.

Our experimental methodology results indicate that our image classification model, in which vague kernels of the 4-th layer were slightly compressed, can achieve the same precision as the state-of-the-art CNN models. The pruning of 5% of network parameters leads only to 1% loss on the traffic sign and traffic light classification precision that would be acceptable if one wants to get better performance in terms of processing capacity or execution time. However, in many cases, especially in deep learning, swapping an existing model for the pruning network elements results in a precision tradeoff. This might be a self-defining goal should the model be precision-oriented in the first place, or should the model be fast processing. Another challenge is how to integrate explanations into an automating driving in order to reduce its complexity and improve the model's performance

6. Conclusions

Many studies have been done to explain the decision-making process of CNNs for classification applications in computer vision. This work shows and experimentally proves that the CNN PRYSTINE model can be compressed using the meaningful perturbations mask. Getting through the proposed methodology, we got an increase in the performance of 17% only with the loss in precision by 6%. However, the compression of around 50% would be leading to low precision of the network and consequently to the low prediction power of traffic signs and traffic lights. The perturbation method can be more generalised and tested on larger data sets and different DNN models for further work. As well as the classification precision might be improved by analysing the CNN kernels comparing correct and error-leading data subsets. Also, compression of the parameters of the different network layers will be valuable to do.

We aim to incorporate the gradient-based explainability methods to identify and understand vague kernels for CNN prediction models in our future work. In addition, it is worth investigating whether the pruning of kernels or parameters is appropriate for network pruning. Finally, we also aim to develop more sophisticated, robust and reliable explainability algorithms to improve the prediction performance of classification models. Such algorithms could pave the way between AI model prediction structures and humans ground-truth knowledge.

Authors Contribution: Conceptualisation K.S., I.N. and K.O. software, K.S.; methodology, K.S., I.N.; validation K.S., I.N.; formal analysis, K.S., I.N; investigation K.S., I.N. and K.O., resources K.S., I.N.; data curation K.S.; writing—original draft preparation K.S., I.N.; writing—review and editing K.S., I.N. and K.O.; visualisation K.S., I.N.; supervision K.O. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The used code is available at:
http://git.edi.lv/kaspars.sudars/prystine_explainable_road_sign_classifier
The used GTRSB - the German Traffic Sign Benchmark database is available at:
https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign
The used LISA Traffic Light Dataset data set is available at:
https://www.kaggle.com/mbornoe/lisa-traffic-light-dataset

Acknowledgement: This research was performed within the "Programmable Systems for Intelligence in Automobiles" (PRYSTINE) project. PRYSTINE has received funding within the Electronic Components and Systems for European Leadership Joint Undertaking (ECSEL-JU) in collaboration with the European Union's H2020 Framework Programme and National Authorities, under grant agreement No. 783190.

Conflict of Interests: The authors declare no conflict of interest.

References

[1] Rolison, J.J.; Regev, S.; Moutari, S.; Feeney. A. What are the factors that contribute to road accidents? An assessment of law enforcement views, ordinary drivers' opinions, and road accident records. *Accident Analysis and Prevention*. **2018**, *115*, 11-24. [CrossRef]

[2] Babić, D.; Fiolić. M.; Darko Babić.; D., Gates.; T. Road Markings and Their Impact on Driver Behaviour and Road Safety: A Systematic Review of Current Findings. *Journal of Advanced Transportation.* **2020**, vol. *2020*, 1-19. [CrossRef] https://doi.org/10.1155/2020/7843743

[3] Godthelp, J. Traffic safety in emerging countries: making roads self-explaining through intelligent support systems. 2019, *26th World Road Congress*. [CrossRef]

[4] Saadna, Y.; Behloul, A. An overview of traffic sign detection and classification methods. *International Journal of Multimedia Information Retrieval*. **2017**, vol. *6*, 193–210. [CrossRef]

[5] Deng, J.; Dong, W.; Socher, R.; Li-Jia, L.; Li, K.; Fei-Fei, L. Imagenet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* **2009**, 248 – 255. [CrossRef]

[6] Ertler, C.; Mislej, J.; Ollmann, T.; Porzi, L.; Neuhold, G.; Kuang, Y. The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale. *arXiv* **2020**, arXiv:1909.04422v2. [CrossRef]

[7] Lindholm, E.; Nickolls, J.; Oberman, S.; Montrym, J. NVIDIA tesla: a unified graphics and computing architecture. *IEEE Micro*. **2008**, 28(2), 39–55. [CrosRef]

[8] Wu, Y.; Liu, Y.; Li, J.; Liu, H.; Hu, X. Traffic Sign Detection based on Convolutional Neural Networks. In International Joint Conference on Neural Networks (IJCNN). **2013**. [CrossRef]

[9] Gu, J.; Wang, Z.; Kuen, J.; Ma L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang.; X. Wang, L.; Wang, G.; Cai, J.; Chen. T. Recent Advances in Convolutional Neural Networks, *arXiv* **2017**, arXiv:1512.07108v6. [CrossRef]

[10] Programmable Systems for Intelligence in Automobiles (PRYSTINE) project. [CrosRef]

[11] Mishra, R.; Gupta, H.P.; Dutta, T. A Survey on Deep Networks Compression: Challenge, Overview, and Solutions. *arXiv* **2020**, arXiv:2010.03954v1. [CrosRef]

[12] Robnik-Šikonja,; M., Bohanec, M. Perturbation-Based Explanations of Prediction Models. *Human and Machine Learning*. **2018**, 59–175. [CrossRef]

[13] Ivanovs, M.; Kadiķis, R.; Ozols, K. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*. **2021**, vol. 150, 228-234. [CrossRef]

[14] Fong, R.; Vedaldi, A. Explanations for Attributing Deep Neural Network Predictions. In W. Samek, G. Montavon, A.Vedaldi, L.K. Hansen, K.-R. Müller (eds.) Explainable AI: Interpreting, Explaining and Visualising Deep Learning, Springer. **2019**, 149–167.

[15] Zeller, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks.*arXiv* **2013**, arXiv:1311.2901v3. [CrossRef]

[16] Zhou B.; Khosla, A.; Lapedriza. A.; Olivia. A.; Torralba, A. Object Detectors Emerging in Deep Scene CNNs, *arXiv* **2015**, arXiv:1412.6865v2. [CrossRef]

[17] Fong, R.; Vedaldi, A. Interpretable Explanations of Black Boxes by Meaningful Perturbation. *Proceedings of the IEEE International Conference on Computer Vision.* **2017**, 3429–3437. [CrossRef]

[18] Fong, R.; Parick, M.; Vedaldi, A. Understanding Deep Networks via Extremal Perturbations and Smooth Masks, *arXiv* **2019,** arXiv:1910.08485v1. [CrossRef]

[19] Petsiuk, V.; Das, A.; Saenlo, K. RISE: Randomised Input Sampling for Explanation of Black-box Models, *arXiv* **2018**, arXiv:1806.07421v3. [CrossRef]

[20] Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations, *arXiv* **2017**, arXiv:1601.08401v3. [CrossRef]

[21] Druml, N.; Macher, G.; Stolz, M.; Armengaud, E.; Watzenig, D.; Steger, C.; Herndl, T.; Eckel, A.; Ryabokon, A.; Hoess, A.; Kumar, S.; Dimitrakopoulos, G.; Roedig H. PRYSTINE-PRogrammable sYSTems for INtelligence in automobilEs. 2018 21st Euromicro Conference on Digital System Design (DSD). IEEE. **2018**. [CrossRef]

[22] Sudars, K. **2021**. PRYSTINE_explainable_road_sign_classifier; GitLab, project ID:547 [CrossRef]

[23] GTRSB - the German Traffic Sign Benchmark. [CrossRef]

[24] LISA Traffic Light Dataset. [CrossRef]