Domain shift problem: activation shaping approach

Liaci Massimiliano - Michela Stefano - Onori Francesco

Automotive Engineering - Autonomous and Connected Vehicle

Politecnico di Torino

Abstract

In the evolving landscape of deep learning, the challenge of facing domain shift remains a critical issue, particularly in the context of image classification tasks. This study introduces a novel approach to mitigate the effects of the problem through the use of activation shaping techniques applied to ResNet-18 (convolutional neural network). By modifying activation maps within the network, we explored two main strategies: the application of random activation maps and the implementation of unsupervised domain adaptation.

The source code implemented in the experiments described in this work is available on Github.

1. Introduction

In machine learning and computer vision, the generalizability of models across different domains is a fundamental concept. Models are often trained on a specific source domain, where they achieve high performance. However, their effectiveness can decrease significantly when applied to a target domain with different characteristics, a problem known as **Domain Shift**. This issue underscores the importance of Domain Adaptation, a process aimed at enabling models to transfer learned knowledge from the source domain to the target domain, thereby improving their performance on data they were not originally trained on.

X	Feature Space
Y	Output Space
x ∈ X	Input Variable
y ∈ Y	Output Variable
D = X, $P(x)$	Domain
T = Y, $P(y\|x)$	Task

Table 1. Notation.

This work adopts the convention of using apex 's' to denote Source Domain and Task, and apex 't' for Target Domain and Task. Thus the **Domain Adaptation Problem** is that $D^s \neq D^t$ and $T^s = T^t$. Testing on the target domain, ideally the network should be able to perform the task with an acceptable value of accuracy. This study focuses on two distinct methodologies aimed at enhancing the adaptability of models across different domains: random shaping of activation maps and unsupervised domain adaptation. Both techniques rely on activation shaping: the activation maps, which are the outputs of convolutions, are modified so that the model learns features that should classify target samples with higher accuracy compared to the model in which no strategy to address domain shift problem is adopted.

The random activation shaping approach does not take advantage of prior knowledge of the target domain, operating under the assumption that its characteristics are unknown during the adaptation process. This technique is based on the hypothesis that randomly removing a portion of the activation maps could increase model performance by making the learned feature representations less domain-specific.

Conversely, unsupervised domain adaptation leverages the prior knowledge of the target domain without access to labeled data in this domain.

This method is particularly relevant in situations where the target domain is accessible for analysis or model training, but labeled data are scarce or absent. Such scenarios are common in many real-world applications, where obtaining comprehensive labeled datasets is often impractical due to the high costs or logistical challenges involved.

1.1. Model and Dataset

This study is based on the ResNet18 model [1], a convolutional neural network widely utilized and recognized for its performance across various artificial vision tasks. ResNet18 is a variant of the ResNet (Residual Networks)

family, characterized by its moderate depth and effectiveness in mitigating the vanishing gradient problem during the training of deep neural networks.

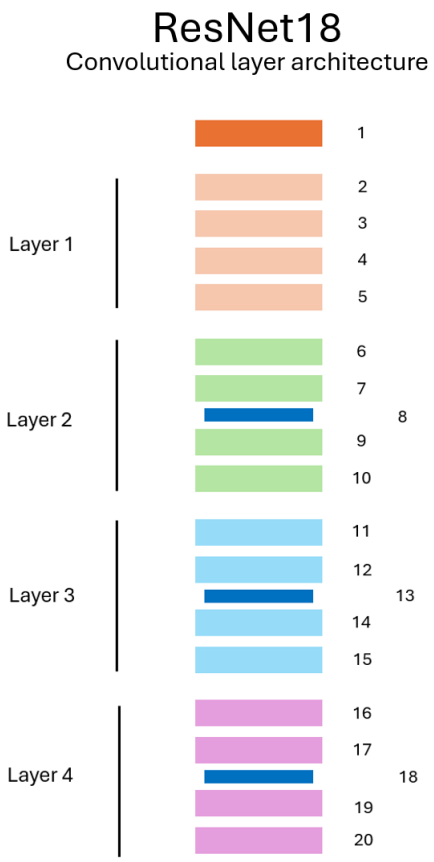


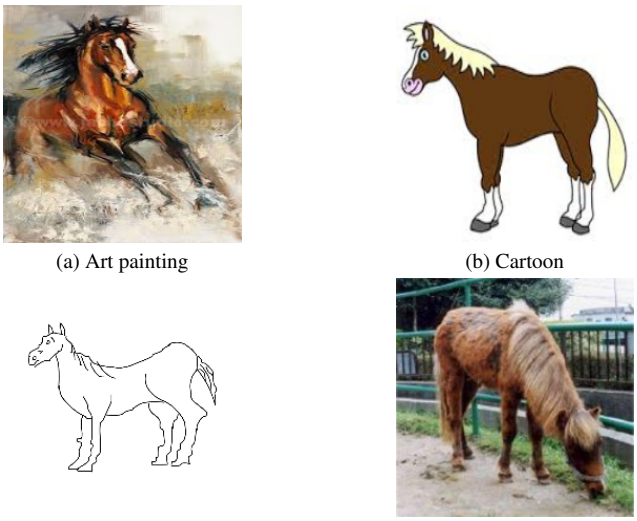
Figure 1. Convolutional layer architecture in ResNet18.

ResNet18 is pre-trained on the extensive ImageNet dataset, a categorized image dataset containing millions of images divided into thousands of classes. This pretraining on ImageNet has been shown to facilitate knowledge transfer across domains, enhancing the network's performance on other image classification tasks.

ResNet18 is composed by 20 convolutional layers organized in 4 macro-layers (except the first convolution). Figure 1 shows that 3 out of 20 convolutions are used for downsampling. From now on any reference to convolutional layers is made according to the enumeration that is made in the figure.

ResNet18 is here trained and tested on a free access dataset known as PACS, usually employed for domain generalization. The PACS dataset consists of images from four different domains: Art Painting, Cartoon, Photo, and Sketch.

In this work, Art Painting is chosen as source domain for training the network, while the other domains (Cartoons, Photo and Sketches) are considered target domains for testing. All images in the PACS dataset are labeled, but during training phase we will use the labels of the source domain only.



(a) Art painting (b) Cartoon

(c) Sketch (d) Photo

Figure 2. PACS image example.

2. Related Works

The approach to domain shift problem adopted in this work is just one among many possible ones. In this section, other studies facing the same issue are reported. The first paper in particular was the foundation for the method used in this study.

Activation Shaping (ASH) method. "Extremely Simple Activation Shaping for Out-of-Distribution Detection" [3], proposes an innovative approach to enhance neural network capability of detecting out-of-distribution (OOD) samples preserving performance (accuracy) on in-distribution (ID) samples classification. The strategy utilized consists of a simple yet effective adjustment of the samples activation maps without the need for retraining or architectural modifications. At inference time, a large portion of the maps is removed (procedure called 'pruning') and the rest is simplified or lightly adjusted.

Adversarial Domain Adaptation. "Domain-Adversarial Training of Neural Networks" [4] presents an innovative domain adaptation technique based on 'domain adversarial training'. The network extracts from input samples features that are then employed to feed two different network

branches: image classifier (labels prediction) and domain classifier. The latter aims at discriminating between source and target samples and is used only during the training phase. The network weights are updated during each training phase minimizing the loss of the label predictor and maximizing the domain classifier one, reversing the gradient flow during back-propagation. The result is that the model learns features that are more domain-invariant, thus mitigating the domain shift issue.

In [3], activation shaping is made by selectively pruning specific zones of the activation maps extracted by the network using a *p-th* percentile as threshold: all the values that are below it are set to zero which is equivalent to keep only the most activated zone of the maps. In this study, we randomly turn off zones of the feature maps independently of the activation level.

3. Methods

This section details the experimental methods employed to face domain shift issue for improving visual recognition performance across diverse domains.

3.1. Produce the baseline

The first step of this study is to produce results without the application of any technique. These accuracy values will be referred to as the 'baseline' and will be used as a benchmark for subsequent experiments.

3.2. Activation maps extraction: forward hook

This section focuses on the implementation of forward hooks [2], a PyTorch tool used to extract and eventually modify activation maps from specific layers within a convolutional neural network (CNN) during the forward pass. This method offers a dynamic and convenient way to access intermediate representations without modifying the network's architecture.

Forward hooks are callable objects attached to layers. Whenever the model forward method is called, the hook function is triggered before proceeding to the next layer. The hook function allows, in this work, to capture the activation maps of specific layers and then manipulate them as required by the experiments. Refer to [2] as documentation on this technique.

3.3. Random activation maps

This method is inspired by the activation shaping technique used in [3] during model testing to enhance out-of-distribution (OOD) detection without impacting in-distribution (ID) sample classification performance. The method implements activation shaping during training under the hypothesis that techniques beneficial for OOD detection can potentially improve model performance in OOD sample classification. The goal is to induce the model to extract features that are less domain-specific and, consequently, less tied to the source domain used for training.

Implementation. Activation shaping is here implemented by applying masks to the binarized activation maps of one or more convolutions in the network. The criterion used for binarization of latters is to replace each value with 1 if positive or with 0 if negative or equal to 0. The application of the mask is done by element-wise product between the activation maps and the masks. The result of this operation is then fed to the next layer of the network.

$$A' = A * M$$

where A' is the new activation map after the shaping, A is the binarized original activation map and M is the random mask. The latter is binary as A. It is generated with a random pattern, controlling as the only parameter that can be varied the probability that an element is a zero (parameter p_0). Consequently, the mask will be approximately composed of a percentage of zeros equal to the probability p_0. A $p_0 = 0$ value corresponds to a mask entirely composed of 1s, which preserves the binarized activation map. Conversely, setting p_0 to 1 generates a mask entirely composed of 0s, which completely cancels the activation map. This value of the parameter was excluded from the experiments. Different single positions or combinations of them are tested for applying activation shaping, excluding the convolutions used for downsampling (numbers 8, 13, 18), where no feature extraction occurs.

Testing and Optimization Phase. The testing phase of this method involves optimizing the classification accuracy as function of two parameters: p_0 and the position/s where to apply activation shaping. Conducting a cross-optimization of both parameters would require a large number of simulations. For instance, testing 10 p_0 values (from 0 to 0.9 with a step of 0.1) and only the odd convolutions (9 convolutions excluding the 13th downsampling convolution) would require 90 experiments. Therefore, the following approach is adopted:

- p_0 variation; activation shaping position fixed after the first convolution in the network: study on the dependency of the accuracy on p_0 only.
- p_0 fixed to the value that performed best in the previous step; activation shaping position variation (single and multiple simultaneous positions tested): study on

the dependency of the accuracy on the position of the activation shaping.

- p_0 variation; activation shaping position fixed to the convolution that performed best in the previous step (or to second best if the first is still convolution 1): this step is necessary to investigate if the optimal value of p_0 obtained by testing activation shaping only on the first convolution is the one that still performs best even on other layers. This way we can hypothesize whether the accuracy depends on the two tested parameters in a separable way.

The optimization phase is conducted on the cartoon domain only. The parameters that perform best on the latter domain are then used to produce results to be compared to the baseline of the other two target domains, sketch, and photo, to assess if they still perform better than their baseline even if optimization is not performed on them.

3.4. Domain adaptation via activation shaping

The activation shaping technique is further exploited to perform unsupervised domain adaptation. Despite all PACS domains being labeled, labels of domains used as targets here are not exploited in any way. The activation maps from the target domains are obtained by passing target samples to the network in inference mode. Subsequently, they are used in the training phase on the source domain as masks to perform activation shaping to convolution outputs. This "injection" of target domain features aims to reduce the gap between the two domains, allowing the network to learn features that are less source-domain specific.

The strategy consists of the following steps:

1. Record activation maps M_t, forwarding x_t through the network (inference mode, no weights update is done).
2. Apply M_t as masks to realize activation shaping when training the network on source domain samples (x_s).
3. Execute the backward pass using Cross-Entropy Loss for optimization of weights and considering the network output z_s and the class label y_s to compute it.

$$A'_i = A_i * M_{t,i}$$

Above the equation of the activation shaping. The subscript 'i' underscores that the masks extracted from the i-th convolutional layer during step 1, have to be positioned on the same i-th convolutional layer during phase 2.

The accuracy evaluation is then computed by forwarding the target domain samples in the trained network.

The accuracy sensitivity on the activation shaping positioning is initially explored using the cartoon target domain. After identifying the most effective position, the possibility to generalize the result to the other domains was evaluated by performing additional tests on sketch and photo domains.

4. Experiments

Here results on Target domains for the baseline model are reported:

Target Domain	Accuracy [%]
Cartoon	54,82
Sketch	35,61
Photo	94,73

Table 2. Baseline accuracy values on Target Domains.

4.1. Random Activation Maps

Regarding this technique, three experiments are conducted, independently analyzing the influence of pruning percentage and mask position.

4.1.1 Pruning influence

In the first experiment, the effectiveness of the percentage of pruning (p_0) to the earliest available convolutional layer in the network architecture is explored. By systematically varying this parameter, the aim is to identify the optimal configuration for maximizing accuracy.

It is important to acknowledge the decision to utilize the first available layer as a starting point for our analysis. While potentially not the optimal layer based solely on performance, this choice was made due to the absence of prior studies analyzing the impact of this technique on different layers.

p_0	Accuracy [%]
0	54,78
0,1	54,82*
0,2	54,69
0,3	55,82*
0,4	57,42
0,5	58
0,6	60,62
0,7	**61,63**
0,8	59,09
0,9	51,88

Table 3. Pruning effect with ASH after the first layer. (* Fitted value from graph).

Fig. 3 and Table 3 displays the achieved accuracy for the Cartoon Domain for varying p_0 values.

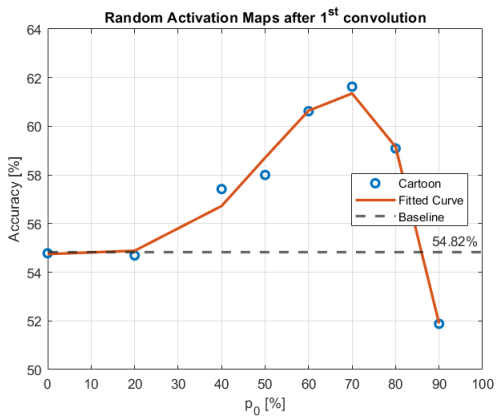


Figure 3. Pruning effect with ASH after the first convolutional layer.

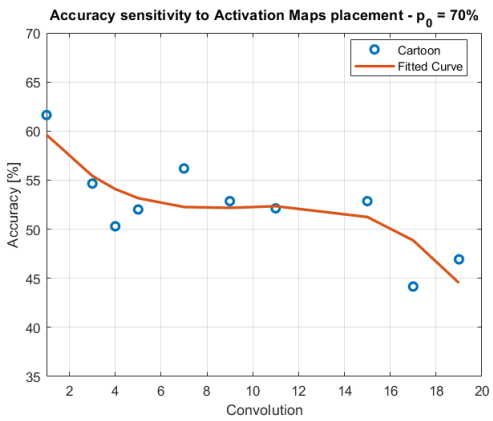


Figure 4. Mask positioning effect.

Furthermore is possible to notice that over a certain percentage of zeros the accuracy abruptly worsen due to a too much invasive deactivation of the activation map extracted.

Note: $p_0 = 0$ *corresponds to a mask composed of only 1, so* $A' = A$.

4.1.2 Mask position influence

From the previous analysis it is obtained that p_0=0.7 is the best pruning percentage for mask placement after the first convolutional layer.

So, in this analysis, it is kept fixed, and the mask will be placed after different convolutional layers (individual and combinations).

Position	Accuracy [%]	Position	Accuracy [%]
1	**61,63**	11	52,13
2	*	12	*
3	54,65	13	**
4	50,3	14	*
5	52,01	15	52,86
6	*	16	*
7	56,19	17	44,16
8	**	18	**
9	52,86	19	46,93
10	*	20	*

Table 4. $p_0 = 0,7$ (* Simulation not performed . ** Simulation not performed because of downsampling layer).

Moving deeper into the network, the features extracted from the convolutions become increasingly specific. Therefore, acting with a mask and deactivating regions of the feature maps results in a deterioration of network performance.

Furthermore, in this analysis, with p0 fixed at 70%, the most effective placement is found to be still after the first convolutional layer. Therefore, the next step of the analysis will involve exploring the influence of the pruning percentage on the second best layer, the seventh.

The investigation is now broadened to include configurations that involve the introduction of the masks after multiple convolutional layers within the network.

- Introduction of masks each three layers, specifically at layers 1, 4, 7, 10, (13), 16, and 19.
- Application of masks to the two layers producing individually the highest accuracy, namely layers 1 and 7.

The results are reported in the table Tab.5

Mask positioning	Cartoon
Baseline	54,82 %
Layer 1	61,63%
Each 3	21,42%
1, 7	48,98%

Table 5. Multiple mask introduction.

4.1.3 Separable dependency test

This step is necessary to investigate if the optimal value of p_0 obtained by testing activation shaping only on the first convolution is the one that still performs best even on other layers. This way we can hypothesize whether the accuracy depends on the two tested parameters in a separable way.

From Fig.4 it is possible to notice that the layer 1 is still the best convolutional layer where to place the mask and all the experiments on it have been already carried out.

Thus, we decided to perform the analysis on the second best layer, the 7^{th}.

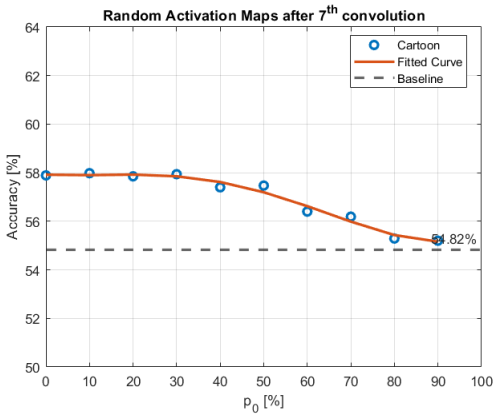


Figure 5. Random Activation Maps after 7^{th} layer.

The fitted curve in Fig.5 shows a different shape compared to the shape of the fitted curve of Fig.3 and the best p_0 value is not anymore 0.7, indicating that the two parameters, p_0 and position, are not separable in their dependency. The effect of adjusting one parameter cannot be fully understood without considering the other.

p_0	Accuracy [%]
0	57.89
0,1	**57.98**
0,2	57.85
0,3	57.94
0,4	57.94
0,5	57.47
0,6	56.4
0,7	56.19
0,8	55.29
0,9	55.2

Table 6. Pruning effect with ASH after the 7^{th} layer.

4.1.4 Other domain test

Upon concluding the experiments within the cartoon domain, subsequent testing was conducted on the other target domains, sketch and photo, utilizing the optimal values determined: $p_0 = 0.7$ with maps applied to the first layer.

It is crucial to recognize that the values identified as optimal for the cartoon domain may not necessarily translate to the best choice for other domains. Anyway, sketch accuracy is still better than baseline even without repeating optimization on this domain.

The outcomes are documented in Table 7; the baseline is also included to facilitate a transparent comparison of the results.

	Cartoon	Sketch	Photo
Baseline	54,82 %	35,61%	94,73%
Random Activation Maps	61,63%	52,30%	80,78%

Table 7. Test on the other Domains.

It is important to note that the performance of the photo domain has deteriorated. The accuracy obtained for the baseline suggests already highly efficient feature extraction, so intervening with masks is corrupting an already very good feature extraction.

4.1.5 Descendent p_0 through Convolutional layers

Starting from the previous results in which it has been noted that a high pruning percentage performs badly towards deeper layers, it is made the hypothesis that previous experiments with multiple masks could have been compromised by the choice of constant p_0 for all the activation shapings performed. It could be meaningful to conduct a new experiment where p_0 decreases gradually moving towards deeper layers of the network.

Referring to Fig.6, we have decided to conduct the analysis involving the first 7 layers of the network since as seen in phase 2 of the experiments, accuracies deteriorate significantly after the 7th layer. For layers 1 and 7, p_0 was chosen by taking the value that had provided the best results for the corresponding layers (0.7 and 0.1 respectively). The intermediate values were fitted with a decreasing exponential function.

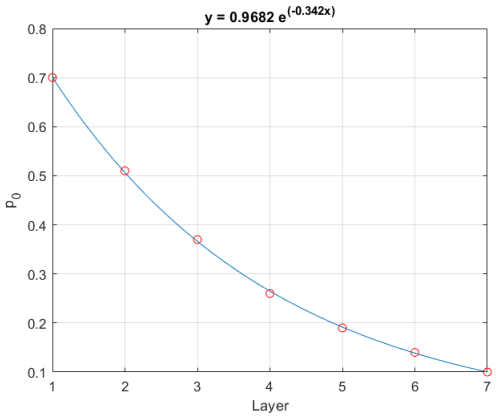


Figure 6. Decreasing Exponential Curve.

In Tab.9 the values of the percentage of pruning for the first seven layers are reported.

Layer	p_0
1	0.7
2	0.51
3	0.37
4	0.26
5	0.19
6	0.14
7	0.1

Table 8. Exponential curve values

Constant p_0	Decreasing p_0
33.28	32.72

Table 9. Constant vs Decreasing p_0 results.

No significant difference occurred between the two configurations. Variable p_0 seems not to be the right solution to enhance multiple layers performances.

4.2. Domain Adaptation: mask position influence

The last experiment aims to investigate the effect of mask placement after different convolutional layers in the Domain Adaptation case.

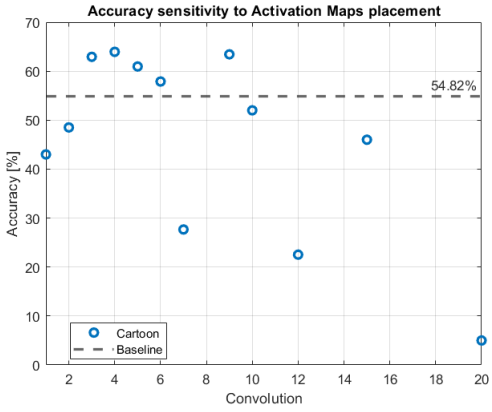


Figure 7. Accuracy sensitivity to Mask placement.

In Fig.7 it can be noticed that a fitting is not useful to describe a general trend of the effect of mask placement. As done for Random Activation Maps downsampling convolutions are excluded from the analysis.

The best results are achieved by placing the mask after convolutional layers 4, 9 and 5. Despite the ineffectiveness of trend fitting operations, it can be hypothesized that as masks are applied after progressively deeper layers, performance deteriorates drastically. This behavior could be attributed to acting on activation maps that correspond to more specific features.

In Table 10 are reported the accuracy values for all the simulations done.

Positioning	Accuracy [%]	Positioning	Accuracy [%]
1	43.0	11	*
2	48.51	12	22.53
3	62.93	13	**
4	**63.95**	14	*
5	60.96	15	46.0
6	57.89	16	*
7	27.67	17	*
8	**	18	**
9	63.44	19	*
10	52.0	20	5.0

Table 10. Mask placement effect (* Simulation not performed . ** Simulation not performed because of downsampling layer).

The investigation is now broadened to include configurations that involve the introduction of the masks after multiple convolutional layer within the network.

- Introduction of masks each three layers, specifically at layers 1, 4, 7, 10, (13), 16, and 19.
- Application of masks to the two layers giving the highest accuracy, namely layers 4 and 9.

The results are presented in Table 11.

Mask positioning	Cartoon	Sketch	Photo
Baseline	54,82 %	35,61%	94,73%
Layer 4	63,95%	50,47%	88,98%
Each 3	15,19%	4,07%	25,87 %
4, 9	58,02%	37,54%	67,90 %

Table 11. Multiple mask introduction.

As mentioned earlier, the optimization process is primarily conducted on the cartoon domain, with subsequent testing on the other. Consequently, conclusions are drawn with a focus on it. While these findings may also apply to other domains, a more thorough analysis is necessary to confirm their validity.

The results show that the highest accuracy is achieved when a mask is applied to only one layer. Introducing masks to an increasing number of layers results in a decline in accuracy. Notably, applying a mask each three layers is particularly detrimental for the performance.

Nevertheless, as done before, some consideration about test on the photo target domain can be discussed: its baseline performance is already highly efficient, achieving an

accuracy of 94.73%. Modifying the activation maps detrimentally affects performance in this domain.

5. Conclusions

In this study the efficacy of activation shaping techniques in addressing the domain shift problem is explored..

By integrating custom activation shaping into the ResNet-18 architecture and utilizing both random activation maps and unsupervised domain adaptation, an analysis to evaluate the impact of these methods on model adaptability is conducted.

Experiments within the cartoon domain provide valuable insights, with the identification of specific layers where activation shaping leads to significant benefits. The application of the same techniques to other target domains, namely sketches and photos, underscores the complexity and domain-specific nature of optimal model tuning. It became evident that strategies driving considerable improvements in one domain may not universally apply to others.

Applying these concepts to the automotive industry, the focus could be shifted for instance, to enhancing the recognition of street signs and pedestrians across different weather conditions (each one representing a different target domain) without compromising recognition capabilities in optimal weather condition (assuming it a source domain). This balance is crucial, especially for SAE Level 4-5 automation vehicles, which should not require driver intervention in any condition. As automation levels increase, the demand for models that maintain high accuracy regardless of the domain shift becomes more important, ensuring safety and reliability in all operating environments.

References

[1] Allena Venkata Sai Abhishek, Venkateswara Rao Gurrala, and Laxman Sahoo. Resnet18 model with sequential layer for computing accuracy on image classification dataset. *ResearchGate*, 2022. Available online at: https://www.researchgate.net/publication/364345322_Resnet18_Model_With_Sequential_Layer_For_Computing_Accuracy_On_Image_Classification_Dataset. 1

[2] Nandita Bhaskhar. Intermediate activations — the forward hook. https://web.stanford.edu/~nanbhas/blog/forward-hooks-pytorch/, 2021. 3

[3] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016. 2