Transient Fault Tolerant Semantic Segmentation for Autonomous Driving

Leonardo Iurada, Niccolò Cavagnero, Fernando Fernandes dos Santos,
Giuseppe Averta, Paolo Rech, Tatiana Tommasi

Transient Fault Tolerant Semantic Segmentation for Autonomous Driving

Leonardo Iurada[1] Niccolò Cavagnero[1] Fernando Fernandes Dos Santos[2]
Giuseppe Averta[1] Paolo Rech[3] Tatiana Tommasi[1]

[1]Politecnico di Torino, Italy [2]Univ Rennes, INRIA, France [3]Università di Trento, Italy

{leonardo.iurada, niccolo.cavagnero, giuseppe.averta, tatiana.tommasi}@polito.it
fernando.fernandes-dos-santos@inria.fr paolo.rech@unitn.it

Abstract

Deep learning models are crucial for autonomous vehicle perception, but their reliability is challenged by algorithmic limitations and hardware faults. We address the latter by examining fault-tolerance in semantic segmentation models. Using established hardware fault models, we evaluate existing hardening techniques both in terms of accuracy and uncertainty and introduce ReLUMax, a novel simple activation function designed to enhance resilience against transient faults. ReLUMax integrates seamlessly into existing architectures without time overhead. Our experiments demonstrate that ReLUMax effectively improves robustness, preserving performance and boosting prediction confidence, thus contributing to the development of reliable autonomous driving systems. Code available at: https://github.com/iurada/neutron-segmentation

1. Introduction

Autonomous vehicles face significant challenges in perceiving and navigating complex environments. Reliable scene recognition models are crucial, especially for Advanced Driver Assistance Systems (ADAS) that must comply with functional safety standards like ISO 26262 [18]. While deep learning has advanced capabilities such as obstacle detection and traffic sign recognition, certification of these components remains a challenge. Recent research has focused on improving algorithmic robustness through domain generalization [31, 38], anomaly detection [32], and open-set recognition [22]. However, hardware robustness is equally critical. Transient hardware faults, often caused by cosmic particles, can result in bit-flip errors [1, 2] that may lead to incorrect predictions and potentially fatal decisions in autonomous vehicles (see Fig. 1). Our work addresses this hardware vulnerability in the context of semantic segmentation, a key task in scene interpretation for autonomous driving. We aim to understand and mitigate the impact of hardware errors on this critical function.

Ideal fault-resilient systems require low-latency and cost-effective strategies. However, current solutions involve expensive hardware or high-cost redundancy [19,39], exemplified by Tesla's Full Self-Driving Chip [36]. Traditional error-correcting code (ECC) focuses on GPU memories rather than functional units [13, 35]. Software-based strategies typically adapt classical techniques to neural networks, introducing significant time overhead [15, 40]. Recent research on computer vision model reliability addresses limited datasets with simplified fault models [27, 37] or relies on reactive post-processing approaches [3], overlooking the model training process.

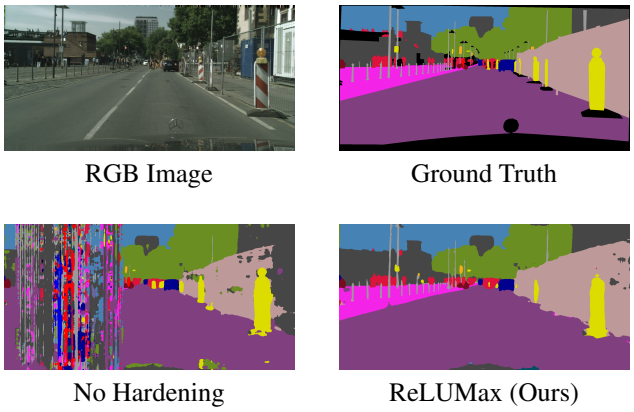

Figure 1. Semantic segmentation models may experience catastrophic output corruption under transient faults, rendering them unsafe for critical applications (No Hardening). To address this limitation, we propose a novel approach for automatically hardening activation functions at training time, without incurring in any additional cost (ReLUMax). Our method ensures robustness against transient faults, mitigating severe corruptions and significantly improving the system's trustworthiness.

With this work, we present to the computer vision community:

- the first fault tolerance analysis on deep learning-based semantic segmentation for autonomous driving. Our study leverages fault models obtained from physical experiments rather than standard synthetic ones.
- a new hardening technique for deep convolutional segmentation models. We introduce the activation function ReLUMax that allows monitoring the training phase and operates corrections at inference time with no latency.

Our experimental evaluation based on a fault injection campaign shows how the proposed solution reduces transient computational errors maintaining an accuracy close to that of the fault-free setting, significantly reducing the number of critical errors and presenting top model confidence.

2. Related Works

Transient Faults and Hardening. Transient faults affect deep learning system reliability, with impact varying by network architecture and processing unit. Software hardening techniques to mitigate these effects are actively researched. Strategies include selective feature map duplication [27], convolution checksums [15, 23, 28, 34], re-execution [11, 24], prediction ensembles [14], specialized pooling layers [21, 34], and value attenuation [8, 21]. Recent approaches designed for object classification combine fault-aware training, ReLU activation clipping, and batch normalization positioning [5]. Only two previous works considered the task of semantic segmentation on fish-eye images. One proposed to improve fault tolerance by calculating activation statistics in a post-processing stage to identify faulty values and apply zero masking on them [3]. The collected statistics capture a late snapshot of the model and do not value the dynamic nature of the training process. The second work [4], adopts a fixed ReLU activation clipping as [5] but observed that it may hinder training convergence and reduce accuracy compared to unmodified models.

Fault Models. Faults can affect any component of deep neural network hardware platforms, with increasing risks as transistors shrink. Fault models represent how faults manifest as incorrect states leading to prediction failures. Common synthetic abstractions include bit-flips in weights, activations, or convolution outputs [3, 4]. Recent studies on neutron beam exposure show faults can corrupt feature maps, affecting entire rows, columns, or blocks of tensors, with magnitudes reaching infinity or NaN [34]. We adopt the strategy from [5], considering random combinations of feature map region variations.

3. Method

Background. One tangible consequence of hardware transient faults is a notable alteration in the range of internal deep network values, often resulting in the emergence of excessively large activations. Existing strategies to tackle this problem include manually setting upper bounds based on the values of the neurons in each layer in the absence of faults [21]. In [3] the authors proposed to collect the distribution of the Average, Minimum, Maximum, and Standard deviation (AMMS) of the activation values for each layer at the end of the training phase. They fix an error detection threshold for each statistic, identifying when it is one standard deviation beyond the minimum or maximal value. They show that if the average and minimum are both out of range, it is possible to reliably identify a fault, and mitigate it by masking values to zero.

We remark that deep neural networks are inherently wired to manage out-of-range activation values thanks to the ReLU functions, thus adding handcrafted procedures is clearly suboptimal. A clipped ReLU activation function was used in [17] to map high-intensity (possibly faulty) activation values to zero. The selected clipping threshold was refined with a dedicated fine-tuning algorithm but could be below the maximum activation value in the training phase, possibly modifying the error-free behavior of the network. In [5] the authors adopted ReLU6 from the literature on efficient deep learning models where the threshold of 6 was chosen to reduce the risk of overflow/underflow [20] and demonstrated to produce the best accuracy-reliability trade-off in case of hardware-permanent faults [17]. Moreover, they exploited fault-aware training via a tailored data augmentation which mimicked the effects of hardware faults to learn patterns that are robust to fault-related noise. Later, [4] discussed how the use of ReLU6 for mitigating faults in segmentation is not without drawbacks.

ReLUMax. We propose to leverage a new version of the ReLU function to improve deep neural network fault resilience for semantic segmentation, while overcoming the limitations of existing approaches. In particular, we introduce ReLUMax, which builds upon the established ReLU-n concept [20], but with a key distinction: it dynamically computes the optimal clipping value for each feature map during the training process. Each ReLUMax activation function stores the observed maximal value (*i.e.* a single floating-point number) from its own output during training and uses it at evaluation time as a trigger to clip activation values to zero.

4. Experiments

In this section we present our experimental analysis to assess the performance of ReLUMax as a hardening solution for fault-resilient semantic segmentation.

4.1. Experimental Setting

Architecture. We use DeepLabV3 [7] on a ResNet-50 backbone [16]. By following standard practices, we start from a pre-trained model obtained on a subset of COCO [25], using only the 20 categories that are present in the Pascal VOC dataset [12].

Datasets. We run the semantic segmentation experiments on GTA5 [33] and Cityscapes [9] datasets, following standard procedures described in [6, 26] and [7, 9], respectively. Both are large-scale datasets for urban scene understanding. The former consists of 24,966 synthetic images 1052×1914 with pixel-perfect annotations. The latter, contains 3,975

	GTA5 [33]						Cityscapes [9]					
	Fault-Free	Fault-Injected					Fault-Free	Fault-Injected				
	mIoU (%)	mIoU (%)	Masked SDCs (%)	No Impact SDCs (%)	Tolerable SDCs (%)	Critical SDCs (%)	mIoU (%)	mIoU (%)	Masked SDCs (%)	No Impact SDCs (%)	Tolerable SDCs (%)	Critical SDCs (%)
No Hardening	78.03 ± 0.22	64.29 ± 0.27	1.12 ± 0.17	0.73 ± 0.15	78.95 ± 2.15	19.20 ± 1.85	**72.62** ± 0.53	51.59 ± 0.67	1.27 ± 0.22	0.60 ± 0.16	75.07 ± 1.86	23.06 ± 1.78
Fault-Aware Training [5]	78.05 ± 0.28	70.78 ± 0.43	1.14 ± 0.17	0.71 ± 0.15	83.72 ± 1.92	14.43 ± 1.63	72.30 ± 0.79	56.14 ± 0.61	1.00 ± 0.19	0.37 ± 0.14	75.66 ± 1.84	22.97 ± 1.79
ReLU6 [4,5]	77.66 ± 0.37	72.60 ± 0.21	0.98 ± 0.16	0.65 ± 0.14	86.21 ± 1.83	12.16 ± 1.42	71.73 ± 1.01	55.12 ± 0.98	0.90 ± 0.18	0.80 ± 0.17	75.20 ± 1.85	23.10 ± 1.77
ReLU6 + Fault-Aware Training [5]	77.83 ± 0.30	73.25 ± 0.47	1.03 ± 0.16	0.65 ± 0.14	87.42 ± 1.74	10.90 ± 1.31	72.14 ± 0.60	58.36 ± 0.73	0.93 ± 0.18	0.80 ± 0.17	76.42 ± 1.82	21.85 ± 1.81
AMMS [3]	78.02 ± 0.25	76.14 ± 0.29	0.95 ± 0.16	0.61 ± 0.14	90.83 ± 1.56	7.61 ± 1.09	72.47 ± 0.53	67.97 ± 0.73	0.97 ± 0.19	0.80 ± 0.17	80.90 ± 1.73	17.33 ± 1.89
ReLUMax (Ours)	**78.06** ± 0.26	**77.48** ± 0.32	0.89 ± 0.15	0.58 ± 0.13	93.27 ± 1.28	**5.26** ± 0.83	72.53 ± 0.72	**70.07** ± 0.83	1.00 ± 0.19	0.60 ± 0.16	85.40 ± 1.65	**13.00** ± 1.95

Table 1. Average fault-free and fault-injected mean Intersection over Union (mIoU), using DeepLabV3 on ResNet-50. Each experiment is repeated three times. We report also the standard deviation. For the Silent Data Corrupts (SDCs) results, we aggregate the number of observed SDCs over the three runs. For Critical SDCs, the lower the better. **Bold** indicates the best results. Underline the second best.

images 1024×2048 with fine-grained annotations of real-world road scenes, comprising up to 30 different categories.

Transient Fault Injection. We inject transient faults using the module from [5]. Errors appear as row or column-wise stripes or localized blocks within the feature maps. Corruption involves multiplying the output tensor with a uniformly sampled random value determining the error magnitude. This fault injection is stochastically applied to random layers during each forward pass.

4.2. Semantic Segmentation Results

We evaluate ReLUMax against five baselines: *No Hardening*, *Fault-Aware Training* [5], *ReLU6* [4, 5], *ReLU6 + Fault-Aware Training* [5], and *AMMS* [3]. We use mean Intersection over Union (mIoU) for evaluation and classify silent data corrupts (SDCs) according to [3, 4] as *Masked*, if no bit-level difference is present in the output logits. *No Impact*, if the predicted pixel-level categories are the same. *Tolerable*, if less than 1% of pixels in the output prediction are affected by the SDCs and no semantic class appears or disappears. They are *Critical* SDCs otherwise. Results are presented in Tab. 1. In fault-free scenarios, hardening methods don't significantly impact performance, though ReLU6 shows a slight decrease. With fault injections, ReLUMax proves most effective, followed by AMMS. Both use similar masking logic, but ReLUMax estimates clipping thresholds during training, generating superior fault estimates.

4.3. Qualitative Effect of Fault Injections

As shown in Fig. 1, the absence of hardening measures leads to substantial corruption from transient faults, posing serious safety risks for autonomous driving. Faults typically generate patterns where entire columns in layer outputs are perturbed. ReLUMax demonstrates a notable ability to mitigate fault influence, providing consistent predictions. Fig. 2 illustrates the worst-case scenario recorded at inference time on Cityscapes. Without hardening, significant performance degradation is observed. Fault-aware training without clipping leads to completely corrupted predictions. ReLU6 activation avoids complete corruption but overestimates the *person* class (in red) in shattered blobs. Combining ReLU6 with fault-aware training or using AMMS doesn't resolve the issue, with the *person* class missing entirely. ReLUMax provides proper localization of persons even in the worst case.

4.4. Uncertainty Analysis of Hardened Models

While mIoU assesses pixel-level segmentation accuracy, it overlooks model confidence, which is crucial in real-world applications where uncertain predictions can have significant consequences. To address this, we analyze model confidence using predictive uncertainty [30], computed via softmax entropy from model outputs. We then evaluate this uncertainty using four metrics proposed in [10, 29, 30]. For the first three metrics we need to decompose the image in patches of size $w \times w$, with $w > 1$ and evaluate on them pixel accuracy and prediction uncertainty. The results are then collected in a confusion matrix containing the number of patches that are accurate and certain n_{ac}, accurate and uncertain n_{au}, inaccurate and certain n_{ic} and inaccurate and uncertain n_{iu}. Finally, we can compute P_{ac} which measures the probability that the model is accurate on its output given that it is confident in its prediction, and P_{ui} which measures the probability that the model is uncertain about its output given that its prediction is wrong. They are respectively defined as

$$P_{ac} : p(\text{accurate} \mid \text{certain}) = \frac{n_{ac}}{n_{ac} + n_{ic}}\,, \tag{1}$$

$$P_{ui} : p(\text{uncertain} \mid \text{inaccurate}) = \frac{n_{iu}}{n_{ic} + n_{iu}}\,. \tag{2}$$

Finally, $PAvPU$ computes the probability of the model being confident on accurate predictions and uncertain on inaccurate ones:

$$PAvPU = \frac{n_{ac} + n_{iu}}{n_{ac} + n_{au} + n_{ic} + n_{iu}}\,. \tag{3}$$

These metrics can be calculated using various uncertainty thresholds, which define the meaning of *certain* for the model. In this regard, we align with [29, 30]: we use $w = 4$ as the window size and 50% as the threshold for defining a patch as accurate (given $w = 4$, at least 9 out of 16 pixels in each patch must be correctly predicted by the model).

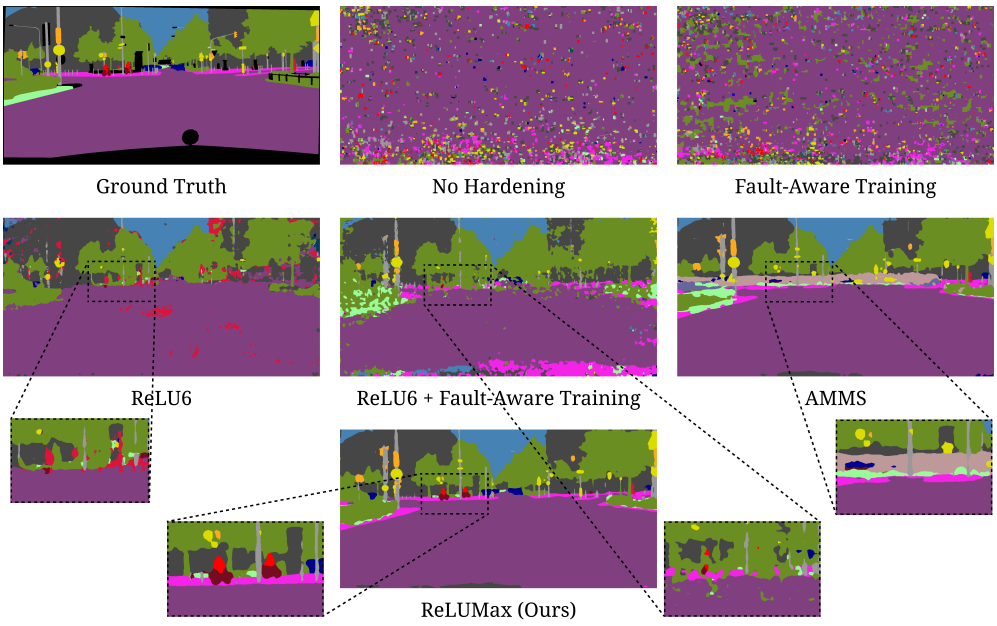

Figure 2. Segmentation maps predicted by the methods under assessment, when simulating injections on the validation split of the Cityscapes dataset. Each color represents a pixel-level annotation of the corresponding semantic class. The choice of the example to visualize is based on the worst recorded mean Intersection over Union (mIoU) when no hardening is introduced. For fairness of comparison, all the methods are injected deterministically in the same way, according to the simulated fault model provided by [5].

	GTA5 [33]								Cityscapes [9]							
	Fault-Free				Fault-Injected				Fault-Free				Fault-Injected			
	P_{ac}	P_{ui}	$PAvPU$	PRR (%)	P_{ac}	P_{ui}	$PAvPU$	PRR (%)	P_{ac}	P_{ui}	$PAvPU$	PRR (%)	P_{ac}	P_{ui}	$PAvPU$	PRR (%)
No Hardening	0.8896	0.6325	0.9977	1.58	0.8974	0.6389	0.9982	1.55	0.8852	0.6102	0.9922	1.27	0.8108	0.5719	0.9496	1.20
Fault-Aware Training [5]	0.8902	0.6378	0.9985	1.63	0.9012	0.6442	0.9988	1.61	0.8830	0.6148	0.9949	1.51	0.8216	0.6080	0.9599	1.45
ReLU6 [4,5]	0.8897	0.6356	0.9981	1.60	0.8998	0.6421	0.9986	1.59	0.8852	0.6128	0.9936	1.47	0.8485	0.6094	0.9867	1.43
ReLU6 + Fault-Aware Training [5]	0.8899	0.6369	0.9983	1.62	0.9005	0.6435	0.9987	1.60	0.8840	0.6090	0.9938	1.39	0.8530	0.5964	0.9852	1.36
AMMS [3]	0.8904	0.6385	0.9986	1.64	0.9018	0.6449	0.9989	1.63	0.8853	0.6105	0.9917	1.44	0.8813	0.6141	0.9955	1.45
ReLUMax (Ours)	**0.8906**	**0.6392**	**0.9988**	**4.85**	**0.9021**	**0.6456**	**0.9990**	**4.80**	**0.8863**	**0.6228**	**0.9993**	**8.28**	**0.8844**	**0.6257**	**0.9996**	**8.26**

Table 2. Uncertainty estimates of the hardening techniques applied to DeepLabV3 on ResNet-50. The results are obtained via predictive uncertainty, computed using softmax entropy. We use a window size of 4x4, the accuracy threshold is set to 50% and the uncertainty threshold is set as the average uncertainty of all pixels over the validation set. **Bold** indicates the best results. Underline the second best.

Moreover, we estimate the uncertainty threshold as the average uncertainty of all pixels over the validation set.

The fourth metric is the *Prediction Rejection Ratio* (PRR) [10]. It is obtained by rejecting samples with low confidence and by computing the accuracy vs the amount of rejected samples (*i.e.* we compute the *Rejection-Accuracy* curves), normalizing the area under the curve by that of an oracle and subtracting a baseline score with randomly sorted samples. A model yielding high values for all four metrics can effectively differentiate between confident, accurate predictions and uncertain, inaccurate ones.

Tab. 2 presents uncertainty results for fault-free inference (first four columns) and under simulated application-level faults (last four columns). Our method consistently outperforms others in uncertainty estimation across all metrics, in both scenarios. Notably, ReLU6 clipping slightly reduces two metrics in fault-free conditions on Cityscapes.

5. Conclusions

We investigated the robustness of semantic segmentation models to transient faults, evaluating existing hardening techniques under realistic fault injection scenarios. Moreover, we propose ReLUMax, a novel activation function that enhances model resilience by identifying acceptable activation ranges during training and clipping high-intensity faulty activations to zero during deployment. Despite its simplicity the proposed solution provides top accuracy results and shows promising performance in terms of maintaining model confidence. Up to our knowledge no previous work assessed uncertainty of hardened models, while we believe it is a crucial aspect to consider, especially in critical application scenarios as autonomous driving.

Future work will extend our study to further architectures (*i.e.* transformer-based) and will involve tests on hardware platforms under neutron beam irradiation by following [5].

Acknowledgements. L.I. acknowledges the grant received from the European Union Next-GenerationEU (Piano Nazionale di Ripresa E Resilienza (PNRR)) DM 351 on Trustworthy AI. T.T. acknowledges the EU project ELSA - European Lighthouse on Secure and Safe AI. This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] Mohammad Hasan Ahmadilivani, Mahdi Taheri, Jaan Raik, Masoud Daneshtalab, and Maksim Jenihhin. A systematic literature review on hardware reliability assessment methods for deep neural networks. *ACM Comput. Surv.*, 56(6), jan 2024. 1

[2] R. Baumann. Soft errors in advanced computer systems. *IEEE Design Test of Computers*, 22(3):258–266, 2005. 1

[3] Stéphane Burel, Adrian Evans, and Lorena Anghel. Improving dnn fault tolerance in semantic segmentation applications. In *DFT*, 2022. 1, 2, 3, 4

[4] Stéphane Burel, Adrian Evans, and Lorena Anghel. Techniques for detecting and masking faults in semantic segmentation applications. *Microelectronics Reliability*, 157:115397, 2024. 2, 3, 4

[5] Niccolò Cavagnero, Fernando Dos Santos, Marco Ciccone, Giuseppe Averta, Tatiana Tommasi, and Paolo Rech. Transient-fault-aware design and training to enhance dnns reliability with zero-overhead. In *IEEE IOLTS*, 2022. 2, 3, 4

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. In *CVPR*, 2017. 2

[8] Zitao Chen, Guanpeng Li, and Karthik Pattabiraman. A low-cost fault corrector for deep neural networks through range restriction. In *DSN*, 2021. 2

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 3, 4

[10] Pau de Jorge, Riccardo Volpi, Philip Torr, and Rogez Gregory. Reliability in semantic segmentation: Are we on the right track? In *CVPR*, 2023. 3, 4

[11] Fernando F. dos Santos, Marcelo Brandalero, Michael B. Sullivan, Pedro M. Basso, Michael Hübner, Luigi Carro, and Paolo Rech. Reduced precision dwc: An efficient hardening strategy for mixed-precision architectures. *IEEE Transactions on Computers*, 71(3):573–586, 2022. 2

[12] Mark Everingham, Ali SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 2

[13] Vinícius Fratin, Daniel Oliveira, Caio Lunardi, Fernando Santos, Gennaro Rodrigues, and Paolo Rech. Code-dependent and architecture-dependent reliability behaviors. In *DSN*, 2018. 1

[14] Zhen Gao, Han Zhang, Xiaohui Wei, Tong Yan, Kangkang Guo, Wenshuo Li, Yu Wang, and Pedro Reviriego. Reliable classification with ensemble convolutional neural networks. In *IEEE DFT*, 2020. 2

[15] Siva Kumar Sastry Hari, Michael Sullivan, Timothy Tsai, and Stephen W Keckler. Making convolutions resilient via algorithm-based error detection techniques. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2021. 1, 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[17] Le-Ha Hoang, Muhammad Abdullah Hanif, and Muhammad Shafique. Ft-clipact: resilience analysis of deep neural networks and improving their fault tolerance using clipped activation. In *DATE*, 2020. 2

[18] ISO. ISO 26262-9:2011 Preview Road Vehicles Functional Safety. https://www.iso.org/standard/51365.html, 2011. 1

[19] Jae-San Kim and Joon-Sung Yang. Dris-3: Deep neural network reliability improvement scheme in 3d die-stacked memory based on fault analysis. In *DAC*, 2019. 1

[20] Alex Krizhevsky. Convolutional deep belief networks on cifar-10. http://www.cs.utoronto.ca/~kriz/conv-cifar10-aug2010.pdf, 2010. 2

[21] Guanpeng Li, Siva Kumar Sastry Hari, Michael Sullivan, Timothy Tsai, Karthik Pattabiraman, Joel Emer, and Stephen W. Keckler. Understanding error propagation in deep learning neural network (dnn) accelerators and applications. In *SC*, 2017. 2

[22] Wuyang Li, Xiaoqing Guo, and Yixuan Yuan. Novel scenes & classes: Towards adaptive open-set object detection. In *ICCV*, 2023. 1

[23] Fabiano Libano, Paolo Rech, and John Brunhaver. Efficient error detection for matrix multiplication with systolic arrays on fpgas. *IEEE Transactions on Computers*, pages 1–14, 2023. 2

[24] F. Libano, B. Wilson, J. Anderson, M. J. Wirthlin, C. Cazzaniga, C. Frost, and P. Rech. Selective hardening for neural networks in fpgas. *IEEE Transactions on Nuclear Science*, 66(1):216–222, 2019. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2

[26] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *CVPR*, 2021. 2

[27] Abdulrahman Mahmoud, Siva Kumar Sastry Hari, Christopher W. Fletcher, Sarita V. Adve, Charbel Sakr, Naresh

Shanbhag, Pavlo Molchanov, Michael B. Sullivan, Timothy Tsai, and Stephen W. Keckler. Optimizing selective protection for cnn resilience. In *ISSRE*, 2021. 1, 2

[28] Sparsh Mittal. A survey on modeling and improving reliability of dnn algorithms and accelerators. *J. Syst. Archit.*, 104(C), 2020. 2

[29] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv:1811.12709*, 2018. 3

[30] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H.S. Torr, and Yarin Gal. Deep deterministic uncertainty: A new simple baseline. In *CVPR*, 2023. 3

[31] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *CVPR*, 2022. 1

[32] S. Rai, F. Cermelli, D. Fontanel, C. Masone, and B. Caputo. Unmasking anomalies in road-scene segmentation. In *ICCV*, 2023. 1

[33] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2, 3, 4

[34] Fernando Fernandes dos Santos, Pedro Foletto Pimenta, Caio Lunardi, Lucas Draghetti, Luigi Carro, David Kaeli, and Paolo Rech. Analyzing and increasing the reliability of convolutional neural networks on gpus. *IEEE Transactions on Reliability*, 68(2):663–677, 2019. 2

[35] Michael B. Sullivan, Nirmal Saxena, Mike O'Connor, Donghyuk Lee, Paul Racunas, Saurabh Hukerikar, Timothy Tsai, Siva Kumar Sastry Hari, and Stephen W. Keckler. *Characterizing And Mitigating Soft Errors in GPU DRAM*. Association for Computing Machinery, 2021. 1

[36] Emil Talpes, Debjit Das Sarma, Ganesh Venkataramanan, Peter Bannon, Bill McGee, Benjamin Floering, Ankit Jalote, Christopher Hsiong, Sahil Arora, Atchyuth Gorti, and Gagandeep S. Sachdev. Compute solution for tesla's full self-driving computer. *IEEE Micro*, 40(2):25–35, 2020. 1

[37] Syed Talal Wasim, Kabila Haile Soboka, Abdulrahman Mahmoud, Salman Khan, David Brooks, and Gu-Yeon Wei. Hardware resilience properties of text-guided image classifiers. In *NeurIPS*, 2023. 1

[38] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger, fewer, & superior: Harnessing vision foundation models for domain generalized semantic segmentation. *CVPR*, 2024. 1

[39] Zheng Xu and Jacob Abraham. Safety design of a convolutional neural network accelerator with error localization and correction. In *ITC*, 2019. 1

[40] Ussama Zahid, Giulio Gambardella, Nicholas J. Fraser, Michaela Blott, and Kees A. Vissers. FAT: training neural networks for reliable inference under hardware faults. *CoRR*, abs/2011.05873, 2020. 1