HARNAS: Neural Architecture Search Jointly Optimizing for Hardware Efficiency and Adversarial Robustness of Convolutional and Capsule Networks

Alberto Marchisio [1] **Vojtech Mrazek** [2] **Andrea Massa** [3] **Beatrice Bussolino** [3] **Maurizio Martina** [3] **Muhammad Shafique** [4]

Abstract

Neural Architecture Search (NAS) methodologies aim at finding efficient Deep Neural Network (DNN) models for a given application under given system constraints. DNNs are compute-intensive as well as vulnerable to adversarial attack threats. To address multiple design objectives, we propose *HARNAS*, a novel NAS framework that jointly optimizes for hardware-efficiency and adversarial-robustness of DNNs executed on specialized hardware accelerators. Besides the traditional convolutional DNNs, *HARNAS* extends the search for complex types of DNNs such as Capsule Networks. For reducing the exploration time, *HARNAS* selects appropriate values of adversarial perturbations to employ in the NAS algorithm. Our evaluations provide a set of Pareto-optimal solutions leveraging the tradeoffs between the above-discussed design objectives.

1 Introduction

Among the Machine Learning models, Deep Neural Networks (DNNs) have shown high performance in a wide variety of applications (Capra et al., 2020)(Grigorescu et al., 2019). Finding an efficient DNN architecture through Neural Architecture Search (NAS) involves a huge number of parameters and typically extremely long exploration time (Pham et al., 2018). The search space becomes even larger when employing NAS algorithms for advanced types of DNNs, such as the Capsule Networks (CapsNets) (Sabour et al., 2017). However, these advancements in DNNs come with multiple design challenges:

1. *High computational complexity:* DNNs require specialized hardware accelerators to be deployed and executed at the edge, where the resources are constrained (Marchisio et al., 2019a).
2. *Security:* DNN models can be fooled by adversarial attacks, which are small and imperceptible perturbations added to the inputs (Shafique et al., 2020). The adversarial robustness is a crucial feature for safety-critical applications (Cheng et al., 2018). Furthermore, integrating security properties during NAS is a challenging, but can enable robust DNN designs (Dave et al., 2022)(Shafique et al., 2021), as compared to the regular DNN design flow.

Hence, the problem is: *how to design advanced DNNs in an energy-efficient and robust way in an automated multi-objective NAS flow?*

1.1 Limitations of State-Of-The-Art and Scientific Challenges

Traditionally, the hardware efficiency of a DNN implemented on a given hardware accelerator is a metric that is typically analyzed after the DNN design, thereby challenging the feasibility of its implementation on resource-constrained IoT devices. The growing interest in hardware efficiency has led to designing Hardware-Aware NAS methodologies (Sekanina, 2021). Also the adversarial robustness of a given DNN is typically investigated once the DNN is already designed. Including the DNN security into the optimization goals of the NAS is a challenging task, because it might lead to a massive search space explosion due to additional factors and extremely time-consuming training and evaluations of numerous candidate solutions. A large pool of adversarial attacks have been proposed in the literature (Yuan et al., 2019), and it is extremely complex to evaluate the adversarial robustness against different attack algorithms. The work in (Guo et al., 2020) proposed a method evaluating the DNN robustness to the PGD attack (Madry et al., 2018) as the optimization goal of the NAS algorithm. *On the contrary, our work performs joint optimizations for the adversarial robustness and hardware efficiency, thereby increasing the complexity of the optimization problem and the training time for evaluating the DNN robustness.* Moreover, it is challenging to model, implement and evaluate the execution on hardware

[1] Institute of Computer Engineering, Technische Universität Wien (TU Wien), Vienna, Austria [2] Faculty of Information Technology, Brno University of Technology, Brno, Czechia [3] Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy [4] eBrain Lab, Division of Engineering, New York University Abu Dhabi, UAE. Correspondence to: Alberto Marchisio <alberto.marchisio@tuwien.ac.at>.

DynN workshop at the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, 2022. Copyright 2022 by the author(s).

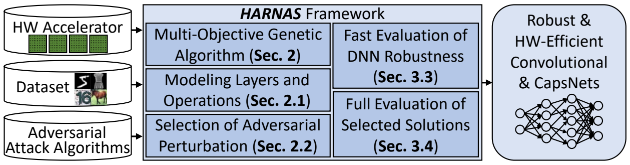
HW Accelerator
Dataset
Adversarial Attack Algorithms
HARNAS Framework
Multi-Objective Genetic Algorithm (Sec. 2)
Modeling Layers and Operations (Sec. 2.1)
Selection of Adversarial Perturbation (Sec. 2.2)
Fast Evaluation of DNN Robustness (Sec. 3.3)
Full Evaluation of Selected Solutions (Sec. 3.4)
Robust & HW-Efficient Convolutional & CapsNets

Figure 1. Overview of our *HARNAS* framework.

devices of different DNN and CapsNet operations (including convolutional layers, fully-connected layers, capsule layers, and dynamic routing) in the NAS design flow.

1.2 Our Novel Contributions

To address the aforementioned challenges, we propose the novel *HARNAS* framework (see Figure 1) that integrates multiple optimization objectives, such as hardware efficiency and adversarial robustness, for advanced types of DNNs and CapsNets. *HARNAS* employs the following key mechanisms:

1. To achieve architectural model flexibility and fast hardware estimation, we deploy analytical models of the layers and operations of DNNs and CapsNets, as well as their mapping and execution on specialized accelerators.
2. For speeding-up the robustness evaluation, we analyze and choose the values of the adversarial perturbations that provide valuable differences when performing the NAS with DNNs subjected to such adversarial perturbations.
3. We develop a specialized evolutionary algorithm, based on the principles of the NSGA-II method (Deb et al., 2002), to perform a multi-objective Pareto-frontier selection, with conjoint optimization for adversarial robustness, energy, memory, and latency of DNNs.
4. To reduce the overall training time, we devise a fast evaluation methodology for DNNs trained for a limited number of epochs, while the Pareto-optimal solutions are evaluated after full-training, to obtain the exact results.

We have implemented our *HARNAS* framework using the TensorFlow library (Abadi et al., 2016), and explored more than 900 DNNs for the MNIST and CIFAR10 datasets. The evaluations are performed on multiple Nvidia V100 GPUs requiring weeks to months of experimentation time.

2 HARNAS Framework

Our evolutionary algorithm-based NAS framework performs a multi-objective search. It searches for inherently robust yet hardware-efficient DNN models by selecting Pareto-optimal candidates in terms of adversarial robustness, energy, latency, and memory footprint. The search space comprises both CNNs and CapsNets. The workflow of our *HARNAS* framework is shown in Figure 2.

The inputs are the hardware accelerator, the adversarial attack algorithm, and the dataset. After modeling analytically the hardware architecture and selecting the values of the adversarial perturbation to employ in the search, the evolutionary algorithm (based on the principles of the NGSA-II genetic algorithm (Deb et al., 2002)) performs an iterative exploration through crossover, mutation, and best DNN candidate selection based on the objectives. To speed up the search, during the evolutionary algorithm, the adversarial robustness is evaluated after training the DNNs with a limited number of epochs, where its number is chosen based on the Pearson Correlation Coefficient (Pearson, 1895). For evaluating the exact robustness results, the set of Pareto-optimal DNN models are fully-trained before measuring their robustness.

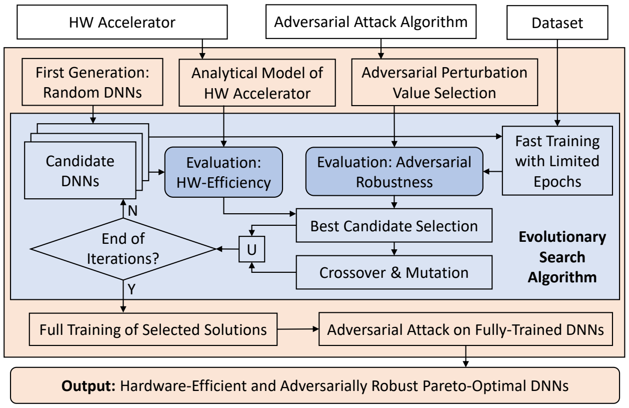
HW Accelerator
Adversarial Attack Algorithm
Dataset
First Generation: Random DNNs
Analytical Model of HW Accelerator
Adversarial Perturbation Value Selection
Candidate DNNs
Evaluation: HW-Efficiency
Evaluation: Adversarial Robustness
Fast Training with Limited Epochs
N
Best Candidate Selection
End of Iterations?
U
Crossover & Mutation
Evolutionary Search Algorithm
Y
Full Training of Selected Solutions
Adversarial Attack on Fully-Trained DNNs
Output: Hardware-Efficient and Adversarially Robust Pareto-Optimal DNNs

Figure 2. Our *HARNAS* framework and its key functionalities.

2.1 Layer and Operation Modeling

The *HARNAS* framework models each layer through a *layer descriptor*, which contains all the parameters for describing the type and sizes of a DNN layer. Any CNN or CapsNet model can be described through multiple layer descriptors, together with information on extra skip connections and resizing of the inputs. To estimate the execution requirements of a DNN model on a specialized DNN hardware accelerator (e.g., CapsAcc (Marchisio et al., 2019b) or TPU (Jouppi et al., 2017)), its underlying hardware characteristics, such as the clock period, the power consumed by the Processing Element (PE) array, the energy consumption and latency required for the memory accessed, etc., must be known. The overall latency, energy consumption, and memory footprint of a DNN model can be computed analytically with the equations in (Marchisio et al., 2020).

2.2 Adversarial Perturbation Value Selection

Since the design space can potentially explode by considering several types and strengths of adversarial noise, the *HARNAS* framework restricts the design space by automatically choosing the values of adversarial perturbations to be employed in the NAS for a given dataset. For each element of the testing dataset, the adversarial example is generated through the PGD algorithm (Madry et al., 2018). Note, here we use the PGD in the discussion, while other adversarial attack algorithms can be integrated into our *HARNAS* framework. When considering the variation of the accuracy w.r.t. the amount of adversarial perturbation (ε), the region in which the slope is highest corresponds to half of the clean accuracy, i.e., $\frac{Acc_0}{2}$. By exploiting this intuition, we select ε_{NAS}, which is the value of adversarial perturbation that provides the closest accuracy to the desired value, which is $\frac{Acc_0}{2}$. The selected value of ε_{NAS} is employed in the *One EPS* search, which optimizes for the robustness against one value of perturbation. Moreover, aiming at covering a wider spectrum of adversarial perturbation ranges, the *Two EPS* search is devised. $\varepsilon_{low} \approx \frac{\varepsilon_{NAS}}{10}$ and $\varepsilon_{high} \approx 3 \cdot \varepsilon_{NAS}$ are computed, and the NAS is conducted by optimizing for the adversarial accuracy with both values.

3 Evaluation of the HARNAS Framework

3.1 Experimental Setup

The tool flow used to implement the *HARNAS* framework and conducting the experiments is summarized as follows. The PGD adversarial attack (Madry et al., 2018) has been implemented with the CleverHans library (Goodfellow et al., 2016). The hardware architecture model has been implemented with the NASCaps library (Marchisio et al., 2020), which is based on the CapsAcc architecture (Marchisio et al., 2019b) synthesized in a 45nm technology node and with a clock period of 3ns using the Synopsys Design Compiler. The DNN training and testing, implemented in TensorFlow (Abadi et al., 2016) have been running on high-end GPU computing nodes equipped with four NVIDIA Tesla V100-SXM2 GPUs. Note that, our experiments were running for 2,000 GPU hours with our fast evaluation method and 8,000 GPU hours for the final training and PGD attack evaluation. Without such exploration time reductions, or by considering more complex optimization problems (e.g., larger datasets or deeper DNN models), the exploration time would have lasted several GPU months.

3.2 Selection of Adversarial Perturbation for the NAS

Following the procedure described in Section 2.2, the Pareto-optimal DNNs of the NASCaps library (Marchisio et al., 2020) have been tested under the PGD attack (Madry et al., 2018), with different values of the adversarial perturbation ε. The results in Fig. 3 show that, as expected, the higher ε is, the lower the DNNs' accuracy drops. The selected values for the NAS are reported in Table 1. The *One EPS* column refers to the search using a single value of ε, while the *Two EPS* column refer to a search conducted with two different values of ε, which are ε_{low} and ε_{high}. Note, a simple dataset like the MNIST requires a relatively high adversarial perturbation to impact the DNN robustness. On the other hand, on a more complex task like the CIFAR10 classification, a smaller perturbation is already sufficient to misclassify a certain set of inputs.

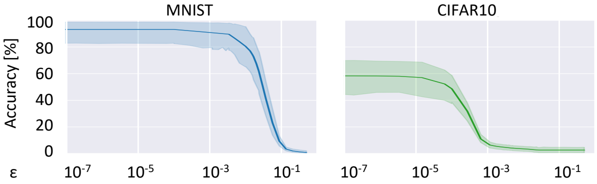

Figure 3. Analysis of the DNN robustness under the PGD attack, with different adversarial perturbation values, for the MNIST and CIFAR10 datasets.

Table 1. Selected values of the adversarial perturbation ε for the NAS, for the MNIST and CIFAR10 datasets. There are also reported the values of ε_{low} and ε_{high} for the *Two EPS* search, which will be used for comparison in Section 3.4.

	Two EPS ε_{low}	One EPS ε	Two EPS ε_{high}
MNIST	3e-3	3e-2	1e-1
CIFAR10	3e-5	3e-4	1e-3

3.3 HARNAS Results with Fast DNN Robustness Evaluation

To reduce the exploration time, our search algorithm trains the DNNs only for a limited number of epochs. The robustness similarity w.r.t. the full-training has been measured through the Pearson Correlation Coefficient (Pearson, 1895), using the methodology described in (Marchisio et al., 2020). The choice of 10 training epochs for the CIFAR10 dataset and 5 epochs for the MNIST dataset leverages the tradeoff between low training time and high correlation.

The results of the *HARNAS* - *One EPS* with fast robustness evaluation are shown in Fig. 4. The earliest generation of the algorithm produces sub-optimal DNN solutions, while most Pareto-optimal solutions are found in the latest generation. Note that, for the *HARNAS* evaluated on the CIFAR10 dataset, the latest generations find DNNs that are less robust to the PGD attack, but still belong to the Pareto-frontier due to the low energy consumption (see pointer ①). Moreover, as highlighted by pointer ②, several candidate DNNs found in the earliest generations are automatically discarded by the Pareto-frontier selection, since they are highly vulnerable to the PGD attack.

3.4 HARNAS Exact Results for Pareto-Optimal DNNs

The Pareto-optimal DNNs selected at the previous stage have been *fully-trained* to evaluate their exact robustness. The DNNs for the MNIST and dataset have been trained for 100 epochs, while the DNNs for the CIFAR10 dataset have been trained for 300 epochs. The results reported in Fig. 5 show tradeoffs between the design objectives. As highlighted by pointer ① in Fig. 5, a Pareto-optimal solution found by the *HARNAS* framework for the CIFAR10 dataset achieves 86.07% accuracy while having an energy

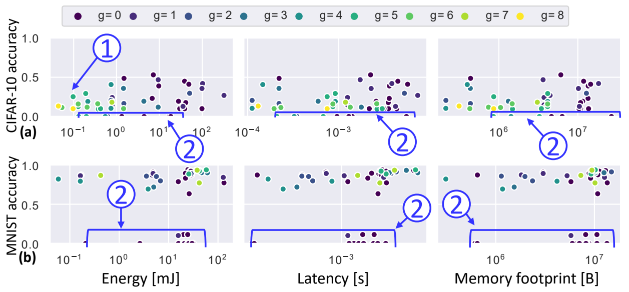

Figure 4. HARNAS' fast evaluation of DNN robustness under PGD attack, showing tradeoffs w.r.t. energy, latency and memory footprint. (a) Results for CIFAR10. (b) Results for MNIST.

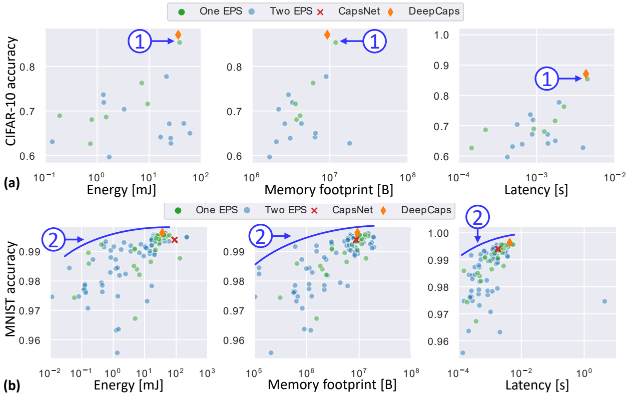

Figure 5. HARNAS' exact robustness evaluation of Pareto-optimal DNNs under the PGD attack, showing tradeoffs w.r.t. hardware-efficiency. (a) Results for CIFAR10. (b) Results for MNIST.

consumption of 38.63 mJ, a memory footprint of 11.85 MiB, and a latency of 4.47 ms. Similarly, the Pareto-optimal DNN search for MNIST covers a wider range of values, leveraging tradeoffs between different objectives (see pointer ②).

The *HARNAS* framework has been compared with other state-of-the-art DNN and CapsNet architectures, and NAS methodologies that include CapsNets in the search space. Fig. 6 shows the comparison between our *HARNAS* framework (*One EPS* setting), NASCaps (Marchisio et al., 2020), CapsNet (Sabour et al., 2017) and DeepCaps (Rajasegaran et al., 2019). For the MNIST dataset, the Pareto-optimal solutions obtained with the *HARNAS* framework are particularly robust for a high range of perturbation ε (see pointer ①). Indeed, the accuracy starts dropping at around one order of magnitude higher ε than NASCaps (see pointer ②). For the CIFAR10 dataset, the *HARNAS* DNNs' behavior is similar to the DeepCaps for low values of ε (see pointer ③), while a Pareto-optimal *HARNAS* solution offer a respectable robustness also with higher adversarial perturbation (see pointer ④).

The evaluation of the *HARNAS* framework with the *Two EPS* setting is shown in Fig. 7. Compared to the *One EPS* setting, the NAS produces different levels of robustness w.r.t. ε for the MNIST dataset (see pointer ① in Fig. 7). However,

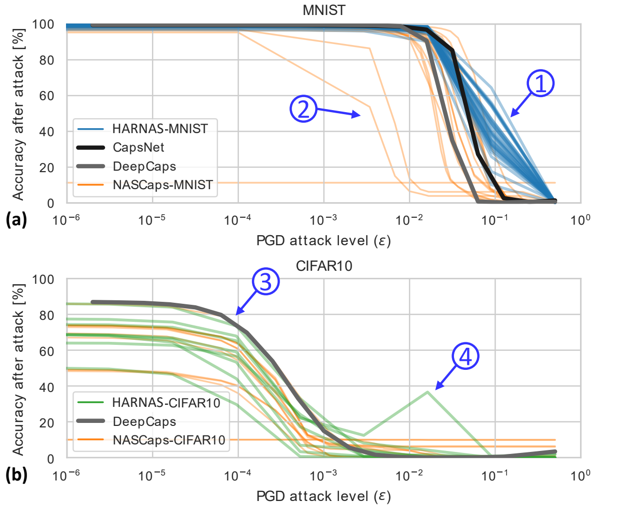

Figure 6. Evaluation of the *HARNAS* framework with the *One EPS* setting, compared to other state-of-the-art architectures and NAS algorithms. (a) Results for MNIST. (b) Results for CIFAR10.

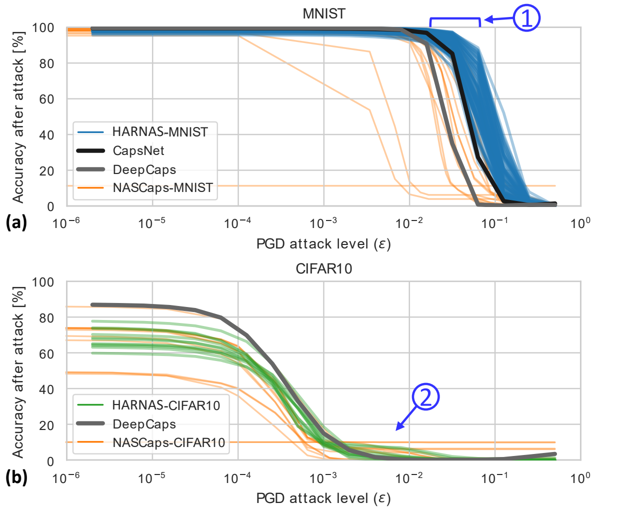

Figure 7. Evaluation of the *HARNAS* framework with the *Two EPS* setting, compared to other state-of-the-art architectures and NAS algorithms. (a) Results for MNIST. (b) Results for CIFAR10.

for the CIFAR10 dataset, the *Two EPS* search leads to worse results than the *One EPS* counterpart (see pointer ②).

4 Conclusion

In this paper, we proposed *HARNAS*, a novel framework for the Neural Architecture Search, jointly optimizing for the hardware efficiency (energy, latency, and memory footprint) and adversarial robustness. Our optimizations reduce the search space and the exploration time. Hence our *HARNAS* framework finds a set of CNNs and CapsNets, which are Pareto-optimal w.r.t. the above-discussed objectives, in a fast fashion. Thanks to our *HARNAS* framework, the deployment of robust DNNs in resource-constrained IoT/edge devices is made possible.

Acknowledgements

This work has been supported in part by the Doctoral College Resilient Embedded Systems, which is run jointly by the TU Wien's Faculty of Informatics and the UAS Technikum Wien. This work was also supported in part by Czech Science Foundation grant GA22-02067S. The computational resources were supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P. A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. Tensorflow: A system for large-scale machine learning. In Keeton, K. and Roscoe, T. (eds.), *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*, pp. 265–283. USENIX Association, 2016. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

Capra, M., Bussolino, B., Marchisio, A., Masera, G., Martina, M., and Shafique, M. Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead. *IEEE Access*, 8:225134–225180, 2020. doi: 10.1109/ACCESS.2020.3039858. URL https://doi.org/10.1109/ACCESS.2020.3039858.

Cheng, C., Diehl, F., Hinz, G., Hamza, Y., Nührenberg, G., Rickert, M., Ruess, H., and Truong-Le, M. Neural networks for safety-critical applications - challenges, experiments and perspectives. In Madsen, J. and Coskun, A. K. (eds.), *2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018*, pp. 1005–1006. IEEE, 2018. doi: 10.23919/DATE.2018.8342158. URL https://doi.org/10.23919/DATE.2018.8342158.

Dave, S., Marchisio, A., Hanif, M. A., Guesmi, A., Shrivastava, A., Alouani, I., and Shafique, M. Special session: Towards an agile design methodology for efficient, reliable, and secure ML systems. *CoRR*, abs/2204.09514, 2022. doi: 10.48550/arXiv.2204.09514. URL https://doi.org/10.48550/arXiv.2204.09514.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002. doi: 10.1109/4235.996017. URL https://doi.org/10.1109/4235.996017.

Goodfellow, I. J., Papernot, N., and McDaniel, P. D. cleverhans v0.1: an adversarial machine learning library. *CoRR*, abs/1610.00768, 2016. URL http://arxiv.org/abs/1610.00768.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 2019. ISSN 1556-4967. doi: 10.1002/rob.21918. URL http://dx.doi.org/10.1002/rob.21918.

Guo, M., Yang, Y., Xu, R., Liu, Z., and Lin, D. When NAS meets robustness: In search of robust architectures against adversarial attacks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 628–637. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00071. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Guo_When_NAS_Meets_Robustness_In_Search_of_Robust_Architectures_Against_CVPR_2020_paper.html.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture, ISCA 2017, Toronto, ON, Canada, June 24-28, 2017*, pp. 1–12. ACM, 2017. doi: 10.1145/3079856.3080246. URL https://doi.org/10.1145/3079856.3080246.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Marchisio, A., Hanif, M. A., Khalid, F., Plastiras, G., Kyrkou, C., Theocharides, T., and Shafique, M. Deep learning for edge computing: Current trends, cross-layer optimizations, and open research challenges. In *2019 IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2019, Miami, FL, USA, July 15-17, 2019*, pp. 553–559. IEEE, 2019a. doi: 10.1109/ISVLSI.2019.00105. URL https://doi.org/10.1109/ISVLSI.2019.00105.

Marchisio, A., Hanif, M. A., and Shafique, M. Capsacc: An efficient hardware accelerator for capsulenets with data reuse. In Teich, J. and Fummi, F. (eds.), *Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019*, pp. 964–967. IEEE, 2019b. doi: 10.23919/DATE.2019.8714922. URL https://doi.org/10.23919/DATE.2019.8714922.

Marchisio, A., Massa, A., Mrazek, V., Bussolino, B., Martina, M., and Shafique, M. Nascaps: A framework for neural architecture search to optimize the accuracy and hardware efficiency of convolutional capsule networks. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2020, San Diego, CA, USA, November 2-5, 2020*, pp. 114:1–114:9. IEEE, 2020. doi: 10.1145/3400302.3415731. URL https://doi.org/10.1145/3400302.3415731.

Pearson, K. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. ISSN 03701662. URL http://www.jstor.org/stable/115794.

Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., and Dean, J. Efficient neural architecture search via parameter sharing. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4092–4101. PMLR, 2018. URL http://proceedings.mlr.press/v80/pham18a.html.

Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., and Rodrigo, R. Deepcaps: Going deeper with capsule networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 10725–10733. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01098. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Rajasegaran_DeepCaps_Going_Deeper_With_Capsule_Networks_CVPR_2019_paper.html.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3856–3866, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2cad8fa47bbef282badbb8de5374b894-Abstract.html.

Sekanina, L. Neural architecture search and hardware accelerator co-search: A survey. *IEEE Access*, 9:151337–151362, 2021. doi: 10.1109/ACCESS.2021.3126685. URL https://doi.org/10.1109/ACCESS.2021.3126685.

Shafique, M., Naseer, M., Theocharides, T., Kyrkou, C., Mutlu, O., Orosa, L., and Choi, J. Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead. *IEEE Des. Test*, 37(2):30–57, 2020. doi: 10.1109/MDAT.2020.2971217. URL https://doi.org/10.1109/MDAT.2020.2971217.

Shafique, M., Marchisio, A., Putra, R. V. W., and Hanif, M. A. Towards energy-efficient and secure edge AI: A cross-layer framework ICCAD special session paper. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2021, Munich, Germany, November 1-4, 2021*, pp. 1–9. IEEE, 2021. doi: 10.1109/ICCAD51958.2021.9643539. URL https://doi.org/10.1109/ICCAD51958.2021.9643539.

Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Networks Learn. Syst.*, 30(9):2805–2824, 2019. doi: 10.1109/TNNLS.2018.2886017. URL https://doi.org/10.1109/TNNLS.2018.2886017.