Multi-Objective Hardware Aware Neural Architecture Search using Hardware Cost Diversity

Nilotpal Sinha, Peyman Rostami, Abd El Rahman Shabayek, Anis Kacem, Djamila Aouada
{nilotpal.sinha, peyman.rostami, abdelrahman.shabayek, anis.kacem, djamila.aouada}@uni.lu
SnT, University of Luxembourg

Abstract

Hardware-aware Neural Architecture Search approaches (HW-NAS) automate the design of deep learning architectures, tailored specifically to a given target hardware platform. Yet, these techniques demand substantial computational resources, primarily due to the expensive process of assessing the performance of identified architectures. To alleviate this problem, a recent direction in the literature has employed representation similarity metric for efficiently evaluating architecture performance. Nonetheless, since it is inherently a single objective method, it requires multiple runs to identify the optimal architecture set satisfying the diverse hardware cost constraints, thereby increasing the search cost. Furthermore, simply converting the single objective into a multi-objective approach results in an under-explored architectural search space. In this study, we propose a Multi-Objective method to address the HW-NAS problem, called MO-HDNAS, to identify the trade-off set of architectures in a single run with low computational cost. This is achieved by optimizing three objectives: maximizing the representation similarity metric, minimizing hardware cost, and maximizing the hardware cost diversity. The third objective, i.e. *hardware cost diversity, is used to facilitate a better exploration of the architecture search space. Experimental results demonstrate the effectiveness of our proposed method in efficiently addressing the HW-NAS problem across six edge devices for the image classification task.*

1. Introduction

Advancements in deep learning systems have brought about a revolutionary impact on various domains, particularly in computer vision [13, 19, 20, 23–25, 33], natural language processing [7, 9, 31], and more. These remarkable achievements were made possible by the creation of meticulously designed architectures that are specifically tailored for individual tasks.

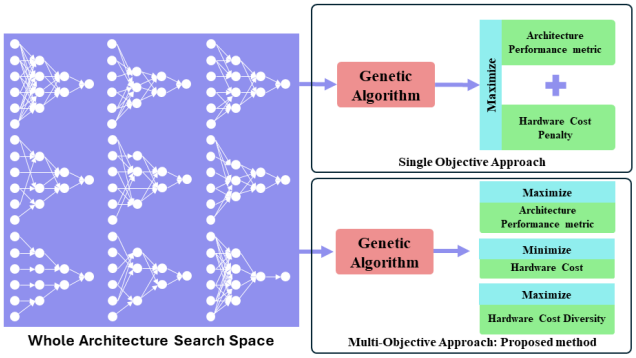

Figure 1. An illustration of the difference between a single objective approach to HW-NAS problem and our proposed method MO-HDNAS.

In response to the growing need for more advanced architectures, researchers have turned their focus towards developing algorithms that can effectively explore the extensive space of neural architectures. These algorithms, collectively known as Neural Architecture Search (NAS) [11, 36, 37], are specifically designed to discover the most optimal architecture for a given task.

The rise in the utilization of edge devices, characterized by low energy consumption, necessitated adaptations to NAS algorithms to incorporate performance considerations from the particular hardware being employed. These customized NAS algorithms are referred to as Hardware-aware Neural Architecture Search (HW-NAS) [1, 2]. While NAS focuses on finding the optimal architecture for a specific task, HW-NAS aims to find architectures with minimal trade-offs between task performance and targeted hardware cost. However, HW-NAS algorithms face a bottleneck due to the extensive time required for evaluating the architecture performance metrics within the search space [22, 37]. This challenge has led to the development of methods that utilize a *supernet-based* solution [3, 4, 17], treating all architectures in the search space as sub-networks of the supernet. While employing this strategy reduces computational

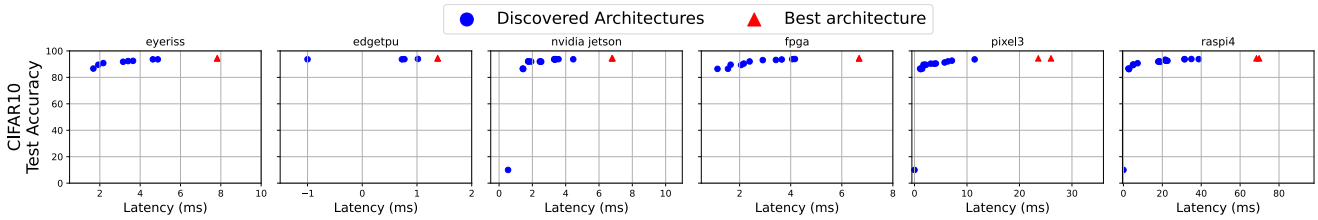

Figure 2. Results of a naive conversion from a single objective to a multi-objective NAS with two objectives: maximize representation similarity and minimizing device latency. It fails to identify the best architecture within the search space. The architecture search is performed in the search space, NAS-Bench-201 [10] on CIFAR10 dataset. More details about the search space are given in Section 4.1.

cost, it compromises architecture search performance due to inaccurate performance estimations by the supernet.

To address the mentioned challenge, [30] has recently proposed the use of a *representation similarity metric* [15, 35], which significantly reduced the search cost while finding the best matching architecture under a single hardware cost constraint. This was achieved by using the *single objective* of maximizing the representation similarity metric with respect to a reference model, while penalizing the search whenever the given architecture constraint is not satisfied (illustrated in Figure 1). However, if multiple different constraints need to be satisfied, the search cost adds up as the algorithm must run multiple times to fulfill each one. Additionally, the naive conversion of the single objective method of [30] to a *multi-objective* one with two objectives (*i.e.* maximizing representation similarity metric and minimizing hardware cost) fails to identify the best architecture. In this regard, Figure 2 illustrates the hardware costs, measured in terms of device latency, of the set of architectures discovered after performing the multi-objective architecture search. It is evident from the figure that the architectures discovered through the architecture search do not exhibit similar performance (test accuracy) to the best architecture found within the search space. This failure is attributed to the high hardware cost of the best architecture, contradicting the second objective aimed at minimizing hardware costs. Note that best architecture in the figure refers to the architecture with the highest accuracy in the search space.

To address these challenges, we propose a Multi-Objective method to address HW-NAS called Multi-Objective Hardware Aware Neural Architecture Search using Hardware Cost Diversity (MO-HDNAS). Our approach aims to identify a set of high-performing architectures with diverse hardware costs in a single run. It achieves this goal by optimizing three objectives (illustrasted in Figure 1): (*1*) Maximizing the representation similarity metric. (*2*) Minimizing hardware cost. (*3*) Maximizing hardware cost diversity. Our contributions can be summarized as follows:

- We generalize the single objective HW-NAS framework proposed in [30] to a multi-objective one in order to address the issue of increased search cost when multiple hardware cost constraints are present.
- We propose a *hardware cost diversity* term aimed at encouraging the consideration of architectures with diverse hardware costs. This allows the search algorithm to explore architectures with higher hardware costs, as high-performing architectures typically tend to have higher hardware cost requirements.
- The robustness of the proposed method is demonstrated on six different edge devices for classification tasks.

2. Related Works

Any NAS method, as described in [12], consists of three key components: *search space*, *search strategy*, and *performance estimation*. The search space generally outlines the potential architectures that can be theoretically represented. Performance estimation involves assessing the expected performance of a neural architecture for a specified task. The search strategy dictates the approach used to explore the defined search space, utilizing architecture performance estimation to find the optimal architecture. It involves techniques such as reinforcement learning (RL)-based methods [21, 37], evolutionary algorithm (EA)-based methods [22, 26–29], and gradient-based methods [17, 34].

Hardware-aware NAS (HW-NAS) is a specialized version of NAS aimed at identifying the optimal architecture tailored for a specific task and target device. HW-NAS typically involves addressing multiple objectives, such as maximizing the architecture performance metric while minimizing the associated hardware cost for the target hardware [1].

Addressing the challenge of multiple objectives can be pursued through two distinct approaches [2]. The first method entails converting the multiple objective problem into a single objective one and solving the latter instead. This can be achieved via rejection sampling [3], which eliminates any architecture that fails to meet the hardware cost constraint during the search process. However, rejection sampling is susceptible to the halting problem, as indicated by [30], particularly when it rejects all candidate architectures for failing to meet a low hardware cost constraint. An alternative solution to rejection sampling involves employ-

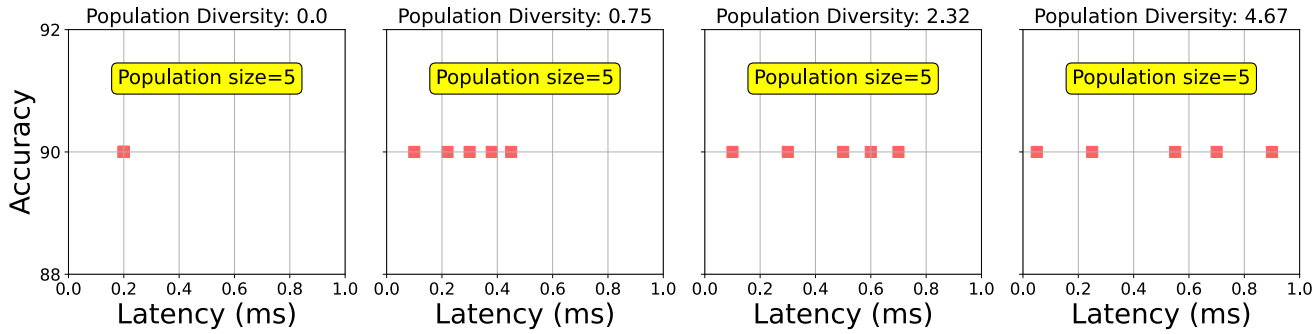

Figure 3. A depiction of the accuracy (y-axis) of all architectures against their respective hardware costs measured in terms of latency (x-axis), in a population size of 5. As the value of diversity increases, the architectures in the population exhibit a spread in hardware costs along the latency axis.

ing a penalty term that reduces the performance metric of an architecture whenever it does not satisfy the hardware cost constraint. Yet, this latter solution suffers from high computational cost. Specifically, when multiple hardware constraints are present, the same single objective problem has to be solved multiple times to accommodate all hardware constraints. The second method to address multiple objectives in HW-NAS employs techniques to identify the pareto optimal solutions [6, 32]. Pareto optimal solutions are those that cannot be improved in one objective without compromising at least one other objective. For instance, improving the accuracy of an architecture may require increasing network parameters, thereby elevating the hardware cost. The pareto approach effectively tackles the elevated search cost issue linked to the single-objective relaxation. Hence, it will be exploited in this work as it provides a set of architectures (pareto optimal set) in a single run, in contrast to multiple runs required by the single objective approach. Our method also ensures that these architectures have diverse hardware costs. This stands in contrast to previous multi-objective methods [6, 32], where the diversity of architecture hardware cost was not considered as one of the objectives.

3. Proposed Method

3.1. Search Method

Our proposed architecture search method employs a metaheuristic optimization technique falling under the category of *genetic algorithms* [14]. These algorithms have demonstrated their effectiveness in addressing the NAS problem [26–28, 30]. They mimic biological adaptation to find optimal solutions in non-differentiable spaces. Starting with an initial population of random neural network architectures, the algorithm iteratively updates/evolves the population, ensuring that the new population $\mathcal{P}$ consists of better performing architectures as compared to previous one. After running the algorithm for a certain number of iterations/generations, the best architecture in the current population is returned as the final solution.

To solve the multi-objective problem of HW-NAS, we employed a popular variant of the genetic algorithm called NSGA-II [8]. It is a well-known *Pareto-based Multi-objective Evolutionary Algorithm (MOEA)*, where selection of individuals is based on *Pareto Efficiency*. In this context, a solution that outperforms others in all objectives is termed "*non-dominated*". Conversely, one that is inferior to others in at least one objective is consistently labeled as "*dominated*". During the selection phase, solutions undergo a sorting process using non-dominated sorting and crowding distance. This technique has been previously employed in the NAS [18, 28] literature, offering a suitable solution for optimizing a neural network architecture based on various objectives.

3.2. Problem Formulation

Let α^* denote a pre-trained *reference model* with a desired performance metric (*e.g.* accuracy for classification task). Also, let $\mathcal{A}$ be the architecture search space in which NAS is performed with α denoting an architecture in the search space. Further, let $\Psi(.)$ denote the function that measures the hardware cost (*e.g.* latency). Formally, the multi-objective hardware-aware architecture search problem can be written as:

$$\begin{aligned} \max_{\alpha \in \mathcal{A}} \quad & \phi(\alpha^*, \alpha), \\ \min \quad & \Psi(\alpha), \\ \max \quad & \chi(\alpha, \mathcal{P}). \end{aligned} \tag{1}$$

This formulation involves solving for three objectives, including:

1. *Maximizing performance similarity metric*, $\phi(\alpha^*, \alpha)$: Finding an architecture α in the search space with similar performance to the reference model α^*. More specifically, the performance similarity metric calculates the

mutual information between hidden layer representation of an architecture and that of the reference model. In other words,

$$\phi(\alpha^*, \alpha) = \sum_{i=1}^{L} I(X^{i*}, X^i), \tag{2}$$

where $X^{1*}, X^{2*}, .., X^{L*}$ and $X^1, X^2, .., X^L$ represent the random variables of feature maps in each layer of α^* and α, respectively. More details are available in [30, 35].

2. *Minimizing hardware cost*, $\Psi(\alpha)$: Finding an architecture α with minimum hardware cost.
3. *Maximizing hardware cost diversity*, $\chi(\alpha, \mathcal{P})$: Maximizing the diversity of the architecture α in terms of the hardware cost, *i.e.* $\chi(\Psi(\alpha, \mathcal{P}))$, as will be discussed in Section 3.3. Note that $\mathcal{P}$ refers to the current generation population of architectures.

3.3. Hardware Cost Diversity

For the current generation population $\mathcal{P}$, the hardware cost diversity term for each architecture α is calculated as

$$\chi(\alpha, \mathcal{P}) = \sum_{\alpha^\dagger \in \mathcal{P}} (\Psi(\alpha) - \Psi(\alpha^\dagger))^2. \tag{3}$$

This formulation measures the difference between the hardware cost of a given architecture α, and those of the remaining architectures $\alpha^\dagger$ in the given population $\mathcal{P}$. Maximizing this term leads to a population characterized by architectures with diverse hardware costs. This is illustrated in Figure 3 which plots the hardware costs of architectures in a population of size five and the impact of the diversity term. This allowed the discovery of architectures with lower latency that preserve the same level of accuracy.

To measure the diversity of the population, we introduce a term called *population diversity*, $\bar{\chi}(\mathcal{P})$, formalized as

$$\bar{\chi}(\mathcal{P}) = \frac{1}{N} \sum_{\alpha \in \mathcal{P}} \chi(\alpha, \mathcal{P}), \tag{4}$$

where N is the population size. It is worth mentioning that $\bar{\chi}(\mathcal{P})$ measures the average hardware cost diversity of architectures within the population.

The leftmost plot in Figure 3 shows the population of architectures with the same hardware cost, consequently resulting in the population diversity term being zero. As we progress to the right on the plots in the figure, we observe an increase in the population diversity term. This ensures that the search algorithm explores the architecture search space, encompassing architectures with varying hardware costs. Please note that the hardware cost used in Figure 3 represents latency. However, the proposed method is agnostic to the specific type of hardware cost utilized for the architecture search.

Algorithm 1: MO-HDNAS

Input: Reference model $\alpha*$, Search space $\mathcal{A}$, Total generations N_{gen}, Population size N_{pop}, training epochs N_{train}
Output: Pareto optimal front of architectures, $P_{optimal}$

1 $\mathcal{P} \leftarrow$ Initialize population for NSGA-II algorithm;
2 $g \leftarrow 0$ (Initialize the generation counter);
3 $archive \leftarrow$ Initialize to empty set;
4 **while** $g \leq N_{gen}$ **do**
5 **for** *each individual architecture* (α) *in* $\mathcal{P}$ **do**
6 $F_{rs} \leftarrow \phi(\alpha^*, \alpha)$ (using Equation 2);
7 $F_{hw} \leftarrow \Psi(\alpha)$;
8 $F_{div} \leftarrow \chi(\alpha, \mathcal{P})$ (using Equation 3);
9 **end**
10 UpdateArchive($\mathcal{P}$, $archive$);
11 $g \leftarrow g + 1$;
12 $\mathcal{P} \leftarrow$ NSGA-II(F_{rs}, F_{hw}, F_{div});
13 **end**

3.4. MO-HDNAS

The pseudo-code of the proposed MO-HDNAS is presented in Algorithm 1. It begins by initializing a population $\mathcal{P}$ consisting of N_{pop} architectures randomly sampled from $\mathcal{A}$. MO-HDNAS iterates for N_{gen} generations. During each generation, the performance of every architecture α (F_{rs}) in the current population, in terms of similarity to reference model, is evaluated in *line* 6 using Equation 2. Next, hardware cost F_{hw} and hardware cost diversity F_{div} of the architectures are calculated in *lines* 7, 8 respectively. Then, the archive is updated in *line* 10 to include the new architectures from the current population. Finally, NSGA-II is used to generate the next generation population in *line* 12. MO-HDNAS returns a pareto optimal front of architectures $P_{optimal}$ (*i.e.* set of possible neural architecture solutions) after N_{gen} generations.

4. Experiments

We adopt the architecture representation introduced in [26] and conduct the architecture search using a single NVIDIA RTX A4000 GPU, with a population size (N_{pop}) set to 20. Following [35], we employ ResNet-20 as the reference model. The representation similarity score is calculated in accordance to the procedure outlined in [30] and the architecture search is performed for 100 generations (N_{gen}).

Further details on the experiments, such as the search space and datasets are presented in Section 4.1 and Section 4.2, respectively. Section 4.3 reports the architecture search performance for six different edge devices, considering various hardware cost settings for each. Finally, an

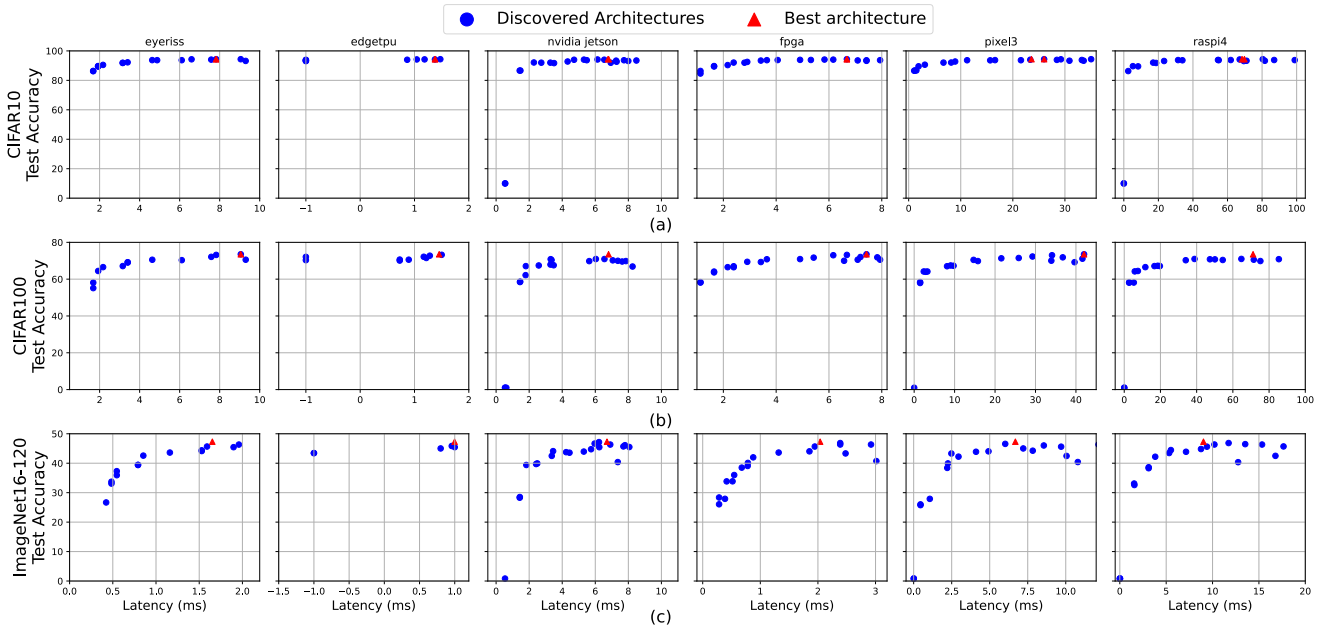

Figure 4. Results of MO-HDNAS for 6 different edge devices performed with only 3 objectives: maximize representation similarity, minimizing device latency and maximizing the hardware cost diversity. (a), (b), (c) show the results for image classification task on CIFAR10, CIFAR100 and ImageNet16-120 respectively.

ablation study is performed on the hardware cost diversity objective in Section 4.4.

4.1. Search Space

The effectiveness of the proposed method is demonstrated on the **NAS-Bench-201** [10] benchmark search space. It provides a unified benchmark for fair comparison of NAS algorithms by providing the results on CIFAR-10, CIFAR-100 and ImageNet16-120 for image classification task. Given that any NAS algorithm aims to search for the type of the operation present between two nodes in a neural architecture, the search space of NAS-Bench-201 includes convolution 3x3, convolution 1x1, max pooling 3x3, skip connection, and none. Note that *none* indicates the absence of any operation between the two nodes. Nevertheless, NAS-Bench-201 lacks information about the hardware cost associated with its architectures. Consequently, we utilize the **HW-NAS-Bench** [16] benchmark. It is an extension of NAS-Bench-201, containing various hardware costs for all architectures in its search space across six edge devices including, *NVIDIA Edge GPU Jetson TX2, Raspberry Pi 4, Edge TPU, Pixel 3, ASIC-Eyeriss, and FPGA*.

4.2. Datasets

We test the effectiveness of the proposed method on three different datasets: *CIFAR-10*, *CIFAR-100*, and *ImageNet-16-120*. *CIFAR-10* consists of 50,000 train and 10,000 test images, categorized into 10 classes. As for the *CIFAR-100*, the number of images in the train and test sets are the same as *CIFAR-10*, but instead coming from 100 classes. On the other hand, the *ImageNet-16-120* [5] is a modified version

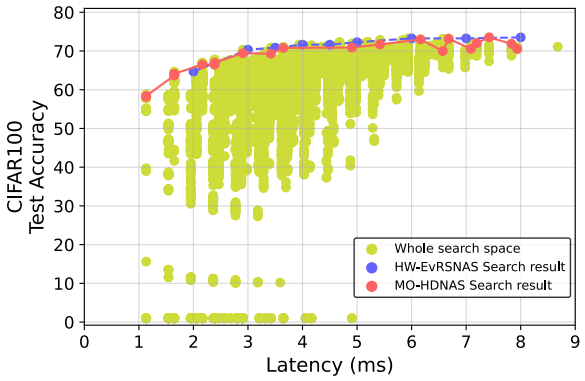

Figure 5. Comparison of architecture search results for FPGA on CIFAR-100 dataset between HW-EvRSNAS [30] and MO-HDNAS.

Methods	Search cost (GPU hours)
HW-EvRSNAS [30]	20.87
MO-HDNAS (Ours)	0.65

Table 1. Search cost comparison of architecture search results for FPGA on CIFAR-100 dataset between HW-EvRSNAS [30] and our method.

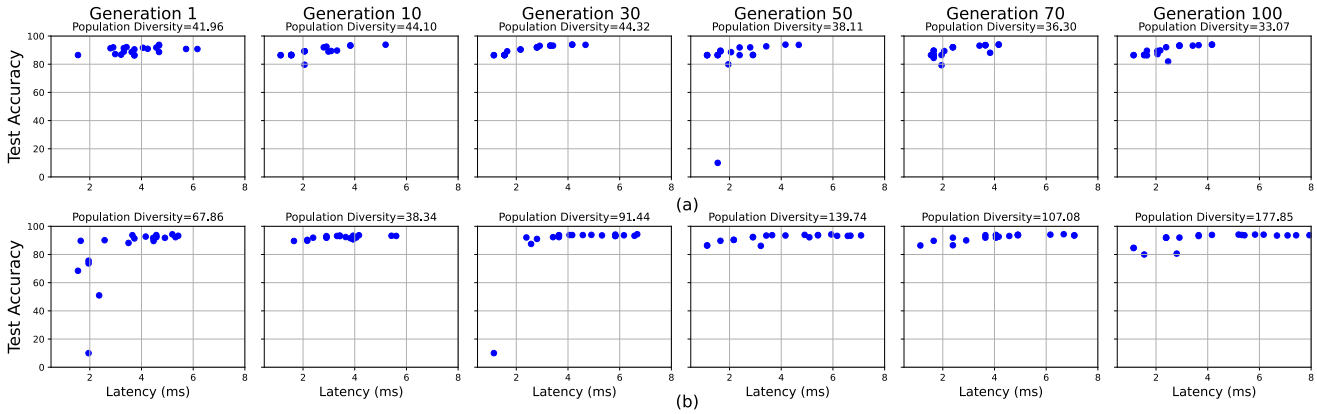

Figure 6. The average hardware costs diversity across different generation population for (a) two objectives (b) three objectives. Note that the search was conducted for the FPGA device using the CIFAR-10 dataset. The x-axis represents the architecture latency on FPGA, while the y-axis depicts the test accuracy of the architecture on CIFAR-10.

of ImageNet containing 120 out of the 1000 total labels, and with each image being downsampled to 16×16 pixels.

4.3. Results

Architecture search results obtained by the proposed MO-HDNAS method are shown in Figure 4. It shows the pareto fronts for the multi-objective architecture search performed on six different edge devices and their hardware cost measured in terms of latency. It is observed that the architectures present in the pareto front of MO-HDNAS are closer to the best architecture. Notably, the pareto front contains architectures with diverse latencies.

To further evaluate the effectiveness of our proposed method, we compare our architecture search results with those of HW-EvRSNAS [30], which treats the HW-NAS problem as a single objective optimization (Figure 1). Results are illustrated in Figure 5, presenting the pareto front discovered by MO-HDNAS and the architecture search results of HW-EvRSNAS under nine different hardware cost constraints. Note that these results are obtained for the image classification task on the CIFAR100 dataset using FPGA. From the figure, it is evident that our method is able to identify a more diverse set of high-performing architectures, ranging from those with low latency to those with high latency.

Furthermore, We compare the search cost of our method with that of HW-EvRSNAS [30] in Table 1. Search costs are reported in terms of GPU hours, indicating the number of hours each method spent to perform the architecture search on a single GPU. These results demonstrate that our method finds the pareto set of architectures at a search cost that is $32\times$ lower than that of HW-EvRSNAS. This is attributed to the fact that HW-EvRSNAS requires nine separate runs to find the optimal architecture for nine different hardware cost constraints. In contrast, our method finds the pareto set of 20 architectures in just a single run.

4.4. Ablation Study

To illustrate the influence of the third objective in the Eq 1 (*i.e.* maximizing hardware cost diversity $\chi(\alpha, \mathcal{P})$), we visualize the *population diversity*, $\bar{\chi}(\mathcal{P})$, of 6 different generations (*1, 10, 30, 50, 70, 100*) in Figure 6. $\bar{\chi}(\mathcal{P})$ measures the average hardware cost diversity for a generation, as showcased in Equation 4. It is computed by taking the average of the $\chi(\alpha, \mathcal{P})$ term across all architectures within that generation's population. Figure 6(a) illustrates the progression of the population diversity term across generations when the objective of maximizing hardware cost diversity is not applied in Equation 1 (*i.e.* only the first two objectives employed). In this case, we observe a decline in population diversity as generations progress. This results in architectures within the population being inclined towards regions with high accuracy and lower hardware costs. Consequently, it hinders the discovery of the best architecture within the search area characterized by high hardware cost.

On the other hand, Figure 6(b) shows the progression of the population diversity term across generations when all three objectives in Equation 1 are utilized in the search process. In this scenario, we observe an increase in population diversity as generation advances. Hence, architectures within the population exhibit diverse hardware costs, spanning from low to high latencies. This enhances the explorability of the search process, facilitating the discovery of high-performing architectures within regions characterized by high hardware cost.

5. Conclusion

In this work, we presented a multi-objective hardware aware neural architecture search method, which performs the ar-

chitecture search with reduced computational cost. This is achieved by searching for architectures with similar internal representation to a reference model, and simultaneously, with minimum hardware cost. Additionally, we introduced a third search objective, hardware cost diversity, to facilitate a better exploration of the architecture search space. The effectiveness of the proposed method is demonstrated on six edge devices for image classification task on three different datasets.

6. Acknowledgement

This work is supported by the Luxembourg National Research Fund (FNR), under the project reference C21/IS/15965298/ELITE.

References

[1] Hadjer Benmeziane, Kaoutar El Maghraoui, Hamza Ouarnoughi, Smail Niar, Martin Wistuba, and Naigang Wang. Hardware-aware neural architecture search: Survey and taxonomy. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4322–4329. International Joint Conferences on Artificial Intelligence Organization, 2021. Survey Track. 1, 2

[2] Hadjer Benmeziane, Kaoutar El Maghraoui, Hamza Ouarnoughi, Smail Niar, Martin Wistuba, and Naigang Wang. A comprehensive survey on hardware-aware neural architecture search. *arXiv preprint arXiv:2101.09336*, 2021. 1, 2

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 1, 2

[4] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019. 1

[5] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 5

[6] Xiangxiang Chu, Bo Zhang, and Ruijun Xu. Multi-objective reinforced evolution in mobile neural architecture search. In *European Conference on Computer Vision*, pages 99–113. Springer, 2020. 3

[7] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011. 1

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002. 3

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[10] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations*, 2020. 2, 5

[11] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *arXiv preprint arXiv:1808.05377*, 2018. 1

[12] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019. 2

[13] Albert Garcia, Mohamed Adel Musallam, Vincent Gaudilliere, Enjie Ghorbel, Kassem Al Ismaeil, Marcos Perez, and Djamila Aouada. Lspnet: A 2d localization-oriented spacecraft pose estimation neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2048–2056, 2021. 1

[14] David E Goldberg. *Genetic algorithms*. pearson education India, 2013. 3

[15] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 2

[16] Chaojian Li, Zhongzhi Yu, Yonggan Fu, Yongan Zhang, Yang Zhao, Haoran You, Qixuan Yu, Yue Wang, and Yingyan (Celine) Lin. {HW}-{nas}-bench: Hardware-aware neural architecture search benchmark. In *International Conference on Learning Representations*, 2021. 5

[17] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. 1, 2

[18] Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. Nsga-net: neural architecture search using multi-objective genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 419–427, 2019. 3

[19] Mohamed Adel Musallam, Vincent Gaudilliere, Enjie Ghorbel, Kassem Al Ismaeil, Marcos Damian Perez, Michel Poucet, and Djamila Aouada. Spacecraft recognition leveraging knowledge of space environment: Simulator, dataset, competition design and analysis. In *2021 IEEE International Conference on Image Processing Challenges (ICIPC)*, pages 11–15, 2021. 1

[20] Marcos Perez, Mohamed Adel Mohamed Ali, Albert Garcia Sanchez, Enjie Ghorbel, Kassem Al Ismaeil, Paul Le Henaff, and Djamila Aouada. Detection & identification of on-orbit objects using machine learning. In *European Conference on Space Debris*, 2021. 1

[21] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4095–4104, Stockholmsmässan, Stockholm Sweden, 2018. PMLR. 2

[22] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4780–4789, 2019. 1, 2

[23] Peyman Rostami, Hojatollah Zamani, Mohammad Fakharzadeh, Arash Amini, and Farokh Marvasti. A

deep learning approach for reconstruction in millimeter-wave imaging systems. *IEEE Transactions on Antennas and Propagation*, 71(1):1180–1184, 2022. 1

[24] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[26] Nilotpal Sinha and Kuan-Wen Chen. Evolving neural architecture using one shot model. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 910–918, 2021. 2, 3, 4

[27] Nilotpal Sinha and Kuan-Wen Chen. Neural architecture search using progressive evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1093–1101, 2022.

[28] Nilotpal Sinha and Kuan-Wen Chen. Novelty driven evolutionary neural architecture search. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 671–674, 2022. 3

[29] Nilotpal Sinha and Kuan-Wen Chen. Neural Architecture Search Using Covariance Matrix Adaptation Evolution Strategy. *Evolutionary Computation*, pages 1–28, 2023. 2

[30] Nilotpal Sinha, Abd El Rahman Shabayek, Anis Kacem, Peyman Rostami, Carl Shneider, and Djamila Aouada. Hardware aware evolutionary neural architecture search using representation similarity metric. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2628–2637, 2024. 2, 3, 4, 5, 6

[31] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 1

[32] Weiqin Ying, Kaijie Zheng, Yu Wu, Junhui Li, and Xin Xu. Neural architecture search using multi-objective evolutionary algorithm based on decomposition. In *Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019, Guangzhou, China, November 16–17, 2019, Revised Selected Papers 11*, pages 143–154. Springer, 2020. 3

[33] Hojatollah Zamani, Peyman Rostami, Arash Amini, and Farokh Marvasti. Elliptical shape recovery from blurred pixels using deep learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2775–2779. IEEE, 2021. 1

[34] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. In *International Conference on Learning Representations*, 2020. 2

[35] Xiawu Zheng, Xiang Fei, Lei Zhang, Chenglin Wu, Fei Chao, Jianzhuang Liu, Wei Zeng, Yonghong Tian, and Rongrong Ji. Neural architecture search with representation mutual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11912–11921, 2022. 2, 4

[36] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017. 1

[37] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018. 1, 2