ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and
Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.

Leveraging performance-based metadata for designing multi-objective NAS strategies for efficient models in Earth Observation

Emre Demir[1,3], Kalifou René Traoré[2,3] and Andrés Camero[3] *

1- Technical University of Munich, School of Computation, Information and Technology
2- Technical University of Munich, Data Science in Earth Observation
3- German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF)

Abstract.
Earth Observational (EO) datasets present challenges that differ from traditional Computer Vision benchmarks often examined by the AutoML community. To assist EO researchers in leveraging AutoML techniques, we offer a NAS benchmark with performance meta-data specifically for an EO context. This dataset not only focuses on resource-efficient models crucial to EO but also includes hardware-based metrics. Moreover, we investigate performance prediction to build a data-centric approach for initializing multi-objective NAS search algorithms.

1 Introduction

Neural Architecture Search (NAS) encompasses a variety of methods designed to automate the challenging process of creating neural network (NN) architectures ([1]). These methods aim to optimize NN architectures for decision-making tasks using various strategies, including random search, bayesian optimization, evolutionary algorithms, and reinforcement learning. ([2]). On the other hand, *Hardware-Aware NAS* (HW-NAS) focuses on customizing models considering hardware constraints, effectively balancing multiple objectives including fitness as well as metrics of model efficiency such as the inference latency ([3]).

Recent AutoML research promotes the use of performance-based metadata databases to manage the computational costs of NAS([4]). NASBenchmarks offer evaluations of large NN search spaces for cost-free prototyping of search methods. NASBench101 and NASBenchNLP([5]) contain tabular performance data of numerous NN configurations. Recent benchmarks make use of surrogate models([6]), allowing for performance predictions across much larger search spaces. While many benchmarks use of Computer Vision (CV) applications, Earth Observational (EO) datasets, with their unique characteristics, require more advanced approaches.

This study explores a dual approach to *HW-NAS* for EO data. It begins with the creation of a specialized *HW-NAS* dataset within the NASBench101 search space, followed by a landscape analysis of this new benchmark. It then examines the effectiveness of using compact surrogate models([1]), to determine initial search points during HW-NAS initialization.

*This work is supported by the Helmholtz AI [grant number: ZT-I-PF-5-01], and a DAAD Doctoral Fellowship.

2 Earth Observation (EO) NAS Benchmark Dataset

EO datasets are essential to foster the development of AI methods for various applications, including climate change monitoring, urban development analysis, and disaster management. Local climate zones (*LCZs*) offer an objective and culture-independent classification system that benefits the analysis of global land-use. In this context, we use the "So2Sat LCZ42" ([7]) dataset to evaluate NN solutions of our database, considering a task of LCZ classification using 10-meter resolution Sentinel-2 imagery.

To select the referred networks, we propose the following sampling strategy: Let us represent the search space defined on NASBench101 ([4]) as a grid, where each *square* is a NN architecture. Then, N=30 random starting points (red color, left figure) from the search space are sampled. Next, a neighbor (one hamming distance) is randomly selected for each point (blue point on the figure in the middle). The process is repeated for each *next* element (e.g., blue points lead to orange ones), creating N random walks. Figure 1 visually describes the process.

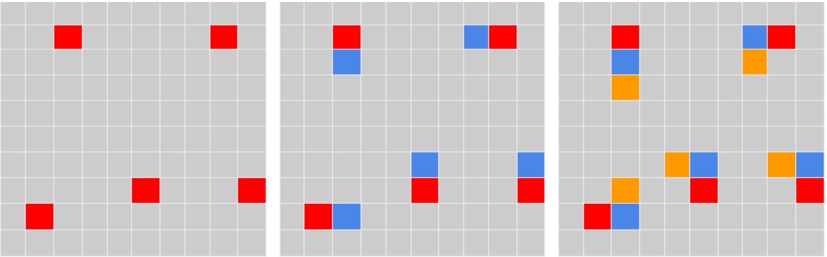

Fig. 1: Deployment of a Random Walk-based sampling strategy over a grid-like search space, for three steps.

With sampling 30 points initially and taking 14 random walk steps on them, we gather 450 unique architectures in total($(30) + (14 * 30) = 450$). The selection of 450 architectures was determined after considering the trade-off between computational resources and the need for a representative sample.

Then, we proceed with evaluating the sample. We measure their training, validation and test accuracies with the LCZ42 So2Sat dataset, as well as their behavior using hardware dependent metrics. These include the average inference time, the standard deviation of the inference time, the architecture size, the number of parameters of the architecture and the *MAC*s ([8]). The fitness is measured at regular intervals of every 2 epochs and until 108 epochs.

For the experiments, each model is trained using a single compute node of the JUWELS Booster partition of the Jülich Supercomputing Centre at Forschungszentrum Jülich. A node comprises four NVIDIA A100 GPUs with 40 GB of virtual memory, and two AMD EPYC Rome 7402 CPUs of 24 cores (2.8 GHz) each. The training and evaluation of the models (train, test, validation) are done in a multi-GPU setting (4 GPUs, Distributed Data Parallel with PyTorch), while the inference (time) is performed using a single GPU.

3 Data-Centric Initialization of NAS

Initialization: We investigate the impact of a data-centric initialization on the performances of a *HW-NAS* search strategy. Since it is a multi-objective optimization problem, we create two surrogate models (XGboost): one for predicting the performance (validation accuracy), another for a hardware related metric (train time). Instead of measuring the real fitness of the encountered solutions, we query their the larger tabular NASBench101 for the relevant performance metadata. We train the surrogates using a very limited amount of data, specifically about 0.5% of the entire search space, which equates to approximately 2,000 architectures.

The next step in the initialization procedure is to use the surrogates to estimate the Pareto-front ([9]) generated using predicted performance of the entire search space. Then, we randomly pick a N non-dominated solutions as starting points for the target search algorithm. This work aims to observe the computing cost benefits of the initialization in terms of number of solution evaluations needed by a MO *HW-NAS* strategy to find the best Pareto-front.

Baseline and Search Evaluation To evaluate the proposed initialization, we use the baseline of *Pareto Local Search*(PLS) ([9]), with a initial population size of $N = 20$ points. We compare the proposed method against a *random initialization*, using 30 different random seeds. During PLS, we prevent duplicate architecture selection by comparing hashes with existing Pareto front solutions. Besides, we compare the initialization baselines considering a budget of encountered solutions (evaluations) by the PLS algorithm. For *PLS*, the solution limit during search depends on the initialization. With a random initialization, the limit is the cumulative sum of training points for the surrogate model and the search limit itself. In contrast, using surrogate initialization only considers the search limit. (Example: 2,000 training points + 500 search limit = 2,500 limit for random init., 500 limit for surrogate init.) All models for the data-centric initialization experiments were trained using the NASLib library ([10]).

4 Results and Experiments

4.1 Landscape Analysis of *EO HW-NAS*

First, we compare the accuracy on both data sets. CIFAR-10 shows balanced accuracy (similar macro/micro acc.), while So2Sat exhibits bias towards the majority class (higher micro). This suggests a more challenging landscape for So2Sat LCZ42. Further analysis reveals class imbalance across train/val/test sets in So2Sat, contributing to the training-test gap. No model or data tuning was performed to maintain a fair EO-NAS benchmark.

Macro accuracy is the average accuracy across all classes, treating each class equally, while micro accuracy is the overall accuracy across all instances, giving more weight to the majority class. A comparison of the average micro and macro classification accuracy across the So2Sat LCZ42 and CIFAR-10 datasets shows that the So2Sat LCZ42 dataset has a macro accuracy of 41.13% and a micro accuracy of 58.67%. In contrast, the CIFAR-10 dataset has both a macro and micro accuracy of 89.62%.

Examining the relationship between the Micro accuracy and the Inference Latency helps visualize the type of MO challenge to be tackled: an ideal baseline retrieves a model (or set of) with as little latency as possible, while providing with a high micro accuracy. This is observed in Figure 3.

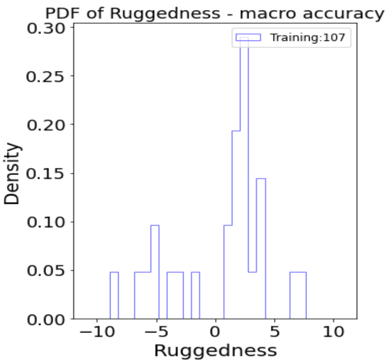

Fig. 2: Ruggedness for Macro Acc.

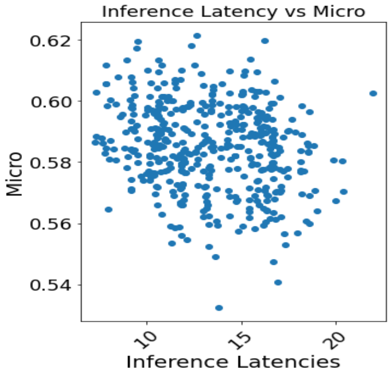

Fig. 3: Latency and MACs

Lastly, the ruggedness which is calculated via the autocorrelation, serves as a significant metric in a landscape analysis, particularly in the context of NAS [11]. Positive ruggedness values indicate less diversity in the search space, while negative values suggest a more challenging and varied space. Understanding the ruggedness aids in optimizing NAS processes. In the generated *EO HW-NAS* dataset, there is a high density for ruggedness values between 0 and 5. Such variations in the distribution suggest the presence of both homogeneous, as well as challenging regions within the search space. This is observed in Figure 2.

4.2 Data Centric Initialization of *HW-NAS*

This study investigates a data-centric initialization for *HW-NAS* within the *NB101* search space. By optimizing cost-effective surrogate models, we construct a performance-driven starting point for *PLS* exploration. We then evaluate the effectiveness of this approach (Pareto Distance Average([12]), Best Accuracy, etc.) by comparing it to a randomly initialized *PLS* with equivalent computational cost (measured by the number of trained models). This comparison allows us to quantify the benefit of the proposed initialization strategy.

To assess the robustness of our findings, the experiment was repeated 30 times with varying random seeds. We focused on comparing "best training time", "best validation accuracy" and "average Pareto distance" across the Pareto fronts from both data-centric and random initialization approaches. Randomly initialized *PLS* achieves higher best validation accuracy (Fig. 5). The narrow metric range (0.935-0.955) may also limit differentiation. Mann-Whitney U-test ([13]) reveals significant differences ($p < 0.05$) in all "best validation accuracy" distributions. A Surrogate initialization led to significantly higher and more sustained exploration diversity based on average Pareto distance (Fig. 4). This suggests early discovery of diverse solutions across the search space, unlike random initialization which might converge prematurely. A Mann-Whitney U-test reveals significant differences ($p < 0.05$) in all "average Pareto distance" distributions.

There was no statistically significant difference observed between the best training times of the compared methods, except for the comparison between surrogate-initialized (500 evaluations) and randomly initialized (2000 evaluations) distributions.

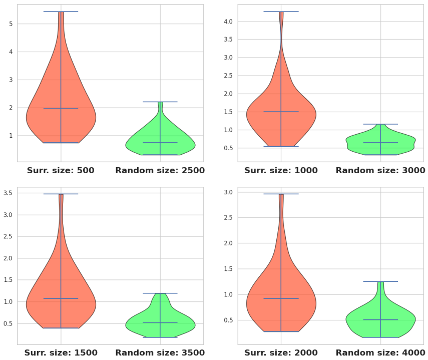

Fig. 4: "Mean Pareto Distance" across 30 separate trials.

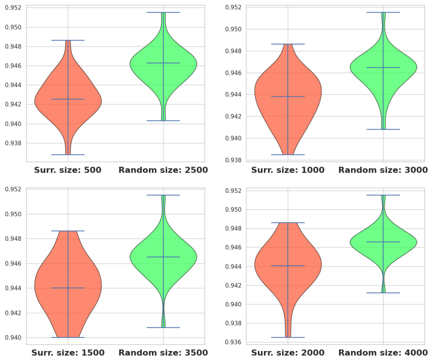

Fig. 5: "Best Validation Accuracy" across 30 separate trials.

5 Conclusion and Future Work

This work introduces an *EO HW-NAS* dataset within the NASBench101 search space and analyzes data-centric initialization for *HW-NAS*.The EO *HW-NAS* dataset empowers Earth Observation *EO* researchers to tailor neural network (NN) designs for their specific data and hardware constraints. This is facilitated by a comprehensive landscape analysis, which explores diverse metrics and yields valuable insights into the *EO* benchmark dataset and its search space. Furthermore, The dataset presents an opportunity for *EO* researchers to develop surrogate models or refine NNs specifically tailored to their unique datasets and hardware configurations.

An analysis of data-centric initialization in HW-NAS on NASBench101 reveals faster training for smaller solutions but marginally lower accuracy compared to random initialization. Notably, this approach generates a more diverse array of solutions. Such architectural diversity, coupled with comparable performance metrics, enhances the robustness and versatility of the resulting models. This diversity facilitates flexible deployment across various hardware platforms, from resource-constrained devices to high-performance systems, thus broadening the applicability and efficacy of the models across diverse computational environments[14].

All code material for these works are publicly available:

- *EO HW-NAS* dataset ([15]).
- Data-Centric initialization of *PLS* ([16]).

References

[1] Colin White colin, Mahmoud Safari, Rhea Sukthanker, Binxin Ru, Thomas Elsken, Arber Zela, Frank Hutter, Colin White, and Debadeepta Dey. Neural architecture search: Insights from 1000 papers. *Arxiv*, 1 2023.

[2] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20:1–21, 2019.

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

[4] Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:12334–12348, 2 2019.

[5] Nikita Klyuchnikov, Ilya Trofimov, Ekaterina Artemova, Mikhail Salnikov, Maxim Fedorov, Alexander Filippov, and Evgeny Burnaev. Nas-bench-nlp: Neural architecture search benchmark for natural language processing. *IEEE Access*, 10:45736–45747, 6 2020.

[6] Arber Zela, Julien Siems, Lukas Zimmer, Jovita Lukasik, Margret Keuper, and Frank Hutter. Surrogate nas benchmarks: Going beyond the limited search spaces of tabular nas benchmarks. *ICLR 2022 - 10th International Conference on Learning Representations*, 8 2020.

[7] Xiao Xiang Zhu, Jingliang Hu, Chunping Qiu, Yilei Shi, Jian Kang, Lichao Mou, Hossein Bagheri, Matthias Haberle, Yuansheng Hua, Rong Huang, Lloyd Hughes, Hao Li, Yao Sun, Guichen Zhang, Shiyao Han, Michael Schmitt, and Yuanyuan Wang. So2sat lcz42: A benchmark data set for the classification of global local climate zones [software and data sets]. *IEEE Geoscience and Remote Sensing Magazine*, 8(3):76–89, 2020.

[8] Ahmed Abdelgawad. Low power multiply accumulate unit (mac) for future wireless sensor networks. *2013 IEEE Sensors Applications Symposium Proceedings*, pages 129–132, 2013.

[9] Jérémie Dubois-Lacoste, Manuel López-Ibáñez, and Thomas Stützle. Pareto local search algorithms for anytime bi-objective optimization. In Jin-Kao Hao and Martin Middendorf, editors, *Evolutionary Computation in Combinatorial Optimization*, pages 206–217, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[10] Michael Ruchte, Arber Zela, Julien Siems, Josif Grabocka, and Frank Hutter. Naslib: A modular and flexible neural architecture search library. `https://github.com/automl/NASLib`, 2020.

[11] Kalifou René Traoré, Andrés Camero, and Xiao Xiang Zhu. Fitness landscape footprint: A framework to compare neural architecture search problems. *CoRR*, abs/2111.01584, 2021.

[12] Arnaud Liefooghe. *Landscape analysis and heuristic search for multi-objective optimization.* Habilitation à diriger des recherches, Université de Lille, June 2022.

[13] Markus Neuhäuser. *Wilcoxon–Mann–Whitney Test*, pages 1656–1658. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[14] Lennart Schneider, Florian Pfisterer, Paul Kent, Juergen Branke, Bernd Bischl, and Janek Thomas. Tackling neural architecture search with quality diversity optimization, 2022.

[15] Demir et al. Eo hw-nas dataset, 2024. https://github.com/emreds/tum-dlr-automl-for-eo.

[16] Demir et al. Data-centric initialization of pls, 2024. https://github.com/emreds/data-centric-nas.