Hardware-Aware Neural Architecture Search for Medical Imaging Applications on Edge Devices

Hadjer Benmeziane[1], Kaoutar El Maghraoui[2], Hamza Ouarnoughi[1] and Smail Niar[1]

[1] UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France

2 IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Introduction

With the transition to electronic health records (EHR) over the last decade, the amount of EHR data has increased exponentially, providing an incredible opportunity to unlock this data with AI to benefit the healthcare system. The particularly sensitive healthcare data requires edge inference and the search for efficient deep learning architectures are particularly needed. Hardware-aware Neural Architecture Search (HW-NAS)[1] has been successfully applied to visual tasks such as image classification and object detection.

We present, MIAS, an adaptable HW-NAS strategy for medical imaging applications. Our approach is based on a flexibly designed supernetwork[2] and efficient multi-objective search. MIAS aims at assisting radiologists and physicians in their daily diagnostics by finding the most-efficient and accurate algorithm for a specific application.

Objectives & Contributions:

- **Introduce MIAS, a flexible and adaptable HW-NAS for medical imaging analysis.**
- **Highlight the significance of data pre-processing in medical AI and integrate an automatic data pre-processing step in MIAS**
- **Present the first NAS benchmark for medical imaging targeting 11 tasks and summarizing performance metrics as well as hardware efficiency.**

MIAS Methodology

MIAS is composed of two main components, shown in Figure 1:

1- Automatic Data Pre-processing: Given a dataset, an input shape, and an output shape, data pre-processing is highly critical, especially for medical data.

- **Imputation**: According to NIFTII and DICOM medical standards, data imputation is applied on the different attribute of the data.
- **Transformation**: Depending on the medical imaging type, i.e., X-rays, a dedicated reshaping and flipping, rotating is achieved.
- **Demographic Analysis**: We analyse the dataset based on demographic attributes such as age and gender. We generate a demographic report which summarizes how general the dataset is.
- **Split:** We split into a balanced training set and a validation set that is proportional to the original dataset.
- **Operator Selection**: We design a dictionary of operators that are suitable for a given input type and output. Based on the input and output shapes and the type of input extracted from the DICOM/NIFTII files, a set of operators is selected for each layer and an over-parameterized supernetwork is designed. The search space is based on a U-Net-like architecture. The upsampling part of the U-Net can be pruned in the case of classification on 2D images.

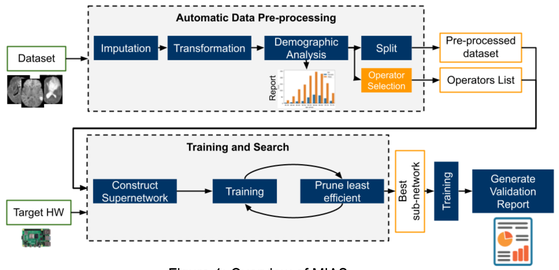


Figure 1: Overview of MIAS.

2- Training & Search: The training of the supernetwork is done alongside the search. During the training, we prune the paths that provide the least task-specific metric/latency trade-off.

- **Supernetwork Construction**: The operator list produced by the first component summarizes the list of possible block structures allowed. Figure 2 shows the general block architecture and the different architecture hyperparameters that build it.

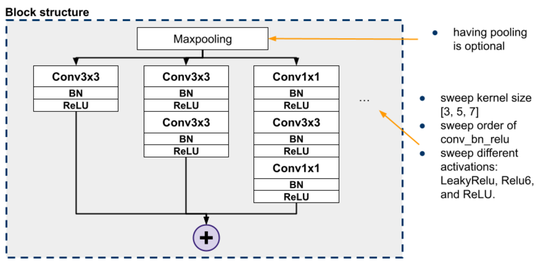


Figure 2: Block architecture in the search space.

Given the operator list, a U-net-like supernetwork architecture is built recursively. Figure 3 shows the U-net-like search space. If a segmentation task is learnt, each block has its equivalent upsampling block. If only a classification/detection is applied, the network is only constructed with a series of down blocks. The number of blocks in the network is also searched.

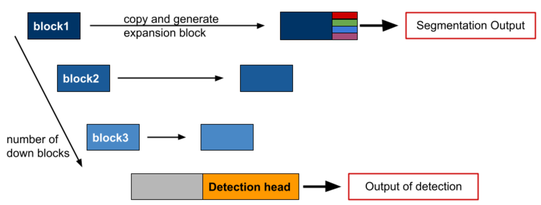


Figure 3: U-Net-like Search Space.

- **Training & Pruning**: The training of the supernetwork is done in a multi-objective way, using this loss function:

$$L = \epsilon_{ce} TSL + \epsilon_{lat} T$$
$$T = \sum_{l} \max_{b}(lat(b))$$

Here, TSL stands for the task-specific loss, ϵ_{ce} and ϵ_{lat} are static weights assigned to the task-specific loss and latency loss. T computes the minimized latency. This sums the latencies of the slowest sampled block in each layer. l is the layer, b is the block and lat is the computed latency of the block on the targeted hardware platform.

For each training iteration, we prune the paths that provide the least task-specific metric/latency trade-off. Note that the task-specific performance metric is curated for medical purposes. In other terms, in a tumour classification scenario, we favour optimizing precision rather than recall to ensure all tumours are detected without any false positives.

- **Validation Report Generation**: Following FDA best practices, our final results are presented in a Validation Plan. The validation plan includes:
  - Intended Use
  - Algorithm Description with a graph of the final model
  - Training dataset information including demographic analysis
  - Validation dataset requirements which defines the ranges of attributes. Outside of these ranges, it is not guaranteed to have a good performance.
  - Performance Results, including task-specific performance and hardware efficiency

Experimentation & Results

Our experiments are conducted on two datasets: brain tumor detection and hippocampus volume estimation. Both datasets come from the Medical Segmentation Decathlon[3]. The targeted hardware platform is Raspberry Pi3.
This edge device is cheap and resource-limited.

The Raspberry Pi 3 Model B is equipped with a Broadcom BCM2837 SoC with a 1.2 GHz quad-core ARM Cortex-A53 CPU, and 1GB RAM, and runs the Raspbian operating system.

Training hyperparameters

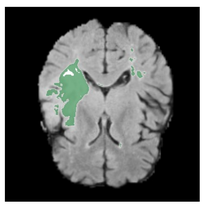
Hyperparameter tuning was achieved during training of the supernetwork. The tuning objective is the task specific loss only. In both trainings, we randomly sample a fixed number of sub-networks that is set to 10.

Hyperparameter	Lr	ϵ_{lat}	ϵ_{ce}	Batch_size	epochs	optim	momentum
Brain-Tumor Detection	0.01	0.3	0.7	32	300	SGD	0.9
Hippocampus Volume Estimation	0.03	0.4	0.6	32	300	SGD	0.9

Overall Resuls

	Brain-tumor detection				Hippocampus Volume Estimation		
	MLP	RF	Nn_unet[4]	Ours	MLP	Nn_unet[4]	Ours
Precision	1.00	0.81	0.89	0.92			
Recall	0.2	0.8	0.83	0.9			
F1-score	0.83	0.8	0.91	0.94			
Dice score	0.73	0.65	0.91	0.943	0.67	0.92	0.954
Latency (ms)*	3.2	2.76	-	3.54	4.54	7.67	3.67

* Latencies were computed on Raspberry Pi3



Brain Tumor Detection

Hippocampus Volume Segmentation

Search Evaluation

We analyze the evolution of sub-network training during training. We use the kendall tau correlation to check if the architectures are correctly ranked. Compared to FairNAS methodology on supernetwork our methodology is more stable and more robust. The ranking correlation is computed by comparing the ranking results during training to the actual ranking of independently trained architectures.

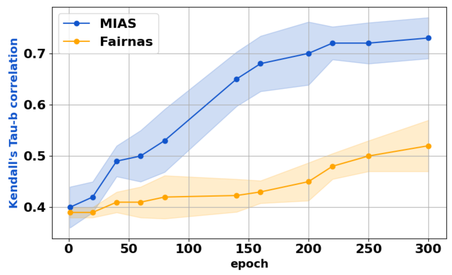


Figure 4: Evolution of the ranking correlation.

Conclusion & Future Works

In summary, our proposed HW-NAS approach for medical imaging applications efficiently searches for deep learning architectures that can be deployed on edge devices while maintaining high accuracy. Our results demonstrate the effectiveness of this approach on brain tumor detection and hippocampus volume estimation tasks, with potential for applications in other medical imaging tasks. Our approach can unlock the potential of EHR data to benefit the healthcare system by providing efficient and accurate medical imaging analysis on edge devices.

The absence of a comprehensive benchmark for Medical HW-NAS is a major bottleneck. To address this limitation, our ongoing research is dedicated to the development of a much-needed benchmark in this domain.
Based on the medical segmentation decathlon, we developed a benchmark called MED-NAS-Bench.

Our benchmark enables:

1. Search for the most efficient and performant architecture on 11 tasks. The benchmark includes task-specific performance such as dice and Jaccard scores as well as efficiency metrics such as latency and energy consumption.
2. Compare HW-NAS for medical segmentation.
3. Provide the benchmark and dataset for multi-task image segmentation. HW-NAS methodologies that claim their multi-task ability should search over 9 tasks and the resulting model should generalize on three unseen tasks.

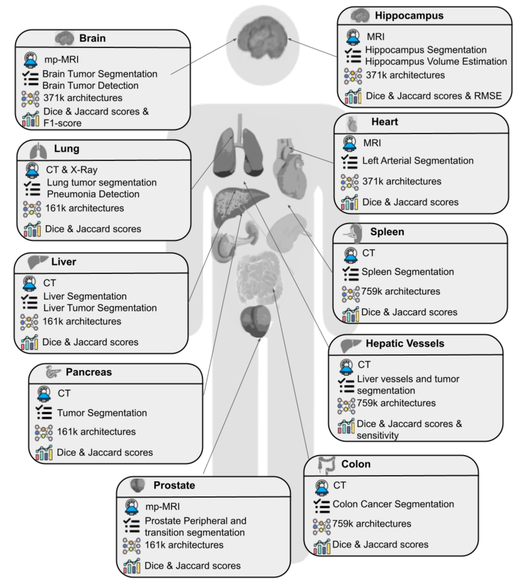


Figure 5 MED-NAS-Bench Overview. This figure is inspired from MSD[3]

REFERENCES

[1] Benmeziane, Hadjer, et al. "Hardware-Aware Neural Architecture Search: Survey and Taxonomy." IJCAI. 2021.

[2] Chen, Xin, et al. "Progressive differentiable architecture search: Bridging the depth gap between search and evaluation." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[3] Antonelli, Michela, et al. "The medical segmentation decathlon." Nature communications 13.1 (2022): 4128.

[4] Isensee, Fabian, et al. "nnu-net: Self-adapting framework for u-net-based medical image segmentation." arXiv preprint arXiv:1809.10486 (2018).