Multimedia Systems (2026) 32:245
https://doi.org/10.1007/s00530-026-02404-9

REGULAR PAPER

Neural architecture search for 3D biomedical image classification

Zeki Kuş[1] · Berna Kiraz[2] · Musa Aydin[3] · Alper Kiraz[4]

Received: 26 December 2025 / Accepted: 3 April 2026

Abstract

3D medical image classification is crucial for improving diagnostic accuracy and treatment planning, but it encounters challenges due to the complexity and variability of volumetric data. While 3D Convolutional Neural Networks offer potential solutions, designing effective architectures is complex and resource-intensive. Neural Architecture Search automates this process, optimizing network designs for specific tasks, thereby improving model performance. This study introduces a novel extension of the PBC-NAS method for 3D medical image classification, aiming to balance prediction accuracy and model complexity. We focus on optimizing neural network architectures using Neural Architecture Search for six different 3D datasets from MedMNIST3D, including OrganMNIST3D, NoduleMNIST3D, FractureMNIST3D, AdrenalMNIST3D, VesselMNIST3D, and SynapseMNIST3D, which are derived from real-world clinical imaging datasets. We have compared our method with state-of-the-art handcrafted networks, AutoML frameworks and recent NAS studies in terms of prediction performance and model complexity. The proposed NAS methods demonstrate superior performance compared to state-of-the-art handcrafted networks and AutoML frameworks. Our proposed model (Ours #3†) achieves the highest average Area Under the Curve (AUC) of 0.915 and accuracy (ACC) of 0.847 (best result across three independent runs), outperforming all handcrafted networks and AutoML frameworks. Compared to other NAS-based methods, all proposed models achieve higher average AUC scores, and it is important to note that they do not rely on data augmentation, pre-processing, or feature selection, unlike the competing NAS methods which do use data augmentation during training. The study also highlights significant reductions in computational complexity, with FLOPs reduced by up to 45.51 times and parameters by up to 211 times compared to ResNet models. An ablation study reveals that while fine-tuning a model optimized for one dataset can achieve competitive results on other datasets, dataset-specific NAS is crucial for optimal performance. Despite this, the ablation results still outperform ResNets and AutoML frameworks in terms of average AUC and ACC. The study concludes that the proposed NAS approach effectively optimizes neural network architectures for complex 3D medical image classification tasks, achieving state-of-the-art performance without data augmentation.

Keywords Neural architecture search · 3D medical image classification · Opposition-based differential evolution

Communicated by Bing-kun Bao.

✉ Zeki Kuş
zkus@fsm.edu.tr

Berna Kiraz
kiraz@itu.edu.tr

Musa Aydin
musa.aydin@samsun.edu.tr

Alper Kiraz
akiraz@ku.edu.tr

[1] Department of Artificial Intelligence and Data Engineering, Fatih Sultan Mehmet Vakif University, İstanbul, Türkiye

[2] Department of Artificial Intelligence and Data Engineering, Istanbul Technical University, İstanbul, Türkiye

[3] Department of Artificial Intelligence and Data Engineering, Samsun University, Samsun, Türkiye

[4] Department of Physics, Department of Electrical and Electronics Engineering, Koç University, İstanbul, Türkiye

Published online: 06 May 2026

1 Introduction

3D medical image classification is an important area in healthcare that uses advanced computer techniques to improve diagnosis accuracy. It primarily uses 3D convolutional neural networks (3D CNNs), which are powerful tools for analyzing medical images [1–4]. These networks can capture detailed spatial information and patterns that 2D methods might miss [5, 6]. Ilesanmi et al. [6] emphasize the increasing importance of 3D CNNs in medical image segmentation and classification, highlighting their capability to identify organs and detect anomalies precisely. Furthermore, the development of large-scale datasets like MedMNISTv2 [7] provides a standardized benchmark for evaluating the performance of machine learning algorithms in both 2D and 3D biomedical image classification tasks.

In recent literature, several studies have introduced handcrafted networks for biomedical image classification [7–14]. Gao Shen et al. [8] presents a novel approach to automate tooth classification using a deep learning model that combines Convolutional Neural Networks (CNNs) and Transformer architectures. This model is designed to handle 3D medical images efficiently, overcoming the high computational demands and large dataset requirements. The authors have validated their approach using both a clinical dataset and the MedMNIST3D dataset. Yang et al. introduce MedMNISTv2 [7], a benchmark dataset collection for 2D and 3D biomedical image classification. It includes six 3D datasets with 9,998 images, covering modalities such as CT scans and electron microscopy for binary and multi-class classification tasks. The authors benchmark deep learning methods (ResNets) and AutoML tools (auto-sklearn [15], AutoKeras [16], Google AutoML Vision) on six different 3D medical image classification datasets. Kiechle et al. [11] explore the potential of Graph Neural Networks (GNNs) as an alternative to the traditional Multi-Layer Perceptrons (MLPs) for classifying 3D medical images. Their study compares the performance of GNNs and MLPs using the MedMNIST3D datasets, demonstrating that GNNs enhance classification performance and significantly improve runtime efficiency. Shin et al. [10] compare AutoML frameworks with handcrafted models for 2D and 3D medical image classification and signal data. They use XGBoost for structured data and EfficientNet with transfer learning for medical image classification. Liu et al. [17] introduce the Feature Pyramid Vision Transformer (FPViT), which combines the ResNet and Vision Transformer (ViT) architectures to address generalization and feature learning challenges in the MedMNIST dataset. By leveraging a feature pyramid structure, FPViT integrates multi-scale feature maps from the layers of ResNet to enhance adaptability and classification accuracy [18]. Schafer et al. [19] propose a novel approach to address the challenge of limited data in biomedical imaging. The authors introduce a multi-task learning strategy that decouples the number of training tasks from memory requirements, allowing for efficient training of a universal biomedical pre-trained model. Lai et al. [20] presents a novel framework that leverages frozen transformer blocks from pre-trained large language models (LLMs) as residual-based encoders for biomedical imaging tasks, achieving significant performance improvements across 2D and 3D datasets. This approach, independent of language-based inputs, performs well on MedMNIST datasets and demonstrates the adaptability of LLMs to visual domains. In MedMNISTv2 [7], the authors have also presented results of AutoML frameworks such as auto-sklearn, AutoKeras, Google AutoML Vision for 3D medical image classification datasets. Shin et al. [10] have extended these frameworks with two additional AutoML frameworks: TPOT [21] and AutoGluon [22]. AutoML frameworks and handcrafted networks often fail to optimize for the unique characteristics of individual datasets, leading to suboptimal performance. Handcrafted networks, while effective in some cases, require significant manual effort and expertise to design, and their fixed architectures may not generalize well across diverse datasets. Furthermore, handcrafted networks and transformer-based models often require significantly more parameters and FLOPs, making them less efficient in resource-constrained medical environments. Additionally, transformer-based models frequently rely on pre-trained architectures, which may not generalize well to 3D medical image classification tasks.

Traditionally, hand-crafted networks require significant expertise and manual effort. Researchers have to try out different configurations to improve performance for specific tasks. Neural architecture search (NAS) is a subfield of AutoML studies that addresses this challenge by using algorithms to explore a wide range of potential architectures for specific tasks [23–25]. NAS studies mainly consist of three key components: the search space, the search method, and the evaluation function. Different search methods are used to automatically explore deep neural network architectures within the predefined search space of a given problem. The main goal is to efficiently identify network architectures that achieve optimal results based on the chosen evaluation function and also reduce the computational complexity within the search space. There are limited studies proposed in the literature for neural architecture search (NAS) in 3D medical image classification [26, 27]. MedPipe [26] presents a novel framework, which integrates the joint search of data augmentation (DA) and NAS for 3D medical image classification. They propose a compact search space that unifies DA and NAS, allowing for their simultaneous exploration and optimization, and evaluated on MedMNIST 3D

datasets. While this joint optimization can enhance performance, it also adds complexity to the search process and restricts the flexibility of the search space, as it must consider both augmentation strategies and network architectures. Additionally, the reliance on data augmentation may not generalize well to datasets with limited variability or where augmentation is less effective. Ali et al. [27] presents an evolutionary approach to NAS for medical image classification, addressing both 2D and 3D datasets. The study introduces a novel method that uses zero-cost proxies to evaluate deep neural networks, significantly reducing the computational cost of the search process. Additionally, it proposes a genetic algorithm-based automatic data augmentation strategy to enhance model generalization and prevent overfitting. The use of zero-cost proxies significantly reduces the computational cost of evaluating architectures, making the method more efficient, although the accuracy of these proxies in estimating performance may be a limitation. Also, the reliance on data augmentation may limit the applicability of the method to datasets where augmentation is less effective.

In our study, we have extended PBC-NAS [28] for 3D medical image classification tasks. In contrast to these methods, our proposed NAS approach focuses solely on optimizing the architecture for 3D medical image classification tasks, without relying on data augmentation or pretrained models. The MedMNIST3D datasets used in our study are derived from real-world clinical imaging datasets, as referenced in the MedMNISTv2 benchmark paper [7], and encompass various imaging modalities. These include CT scans (e.g., OrganMNIST3D [29], NoduleMNIST3D [30], AdrenalMNIST3D, FractureMNIST3D [31]), MRA (e.g., VesselMNIST3D [32]), and electron microscopy (e.g., SynapseMNIST3D), representing common diagnostic techniques in clinical practice. They cover multiple anatomical regions and conditions, including organ classification, lung nodule detection, rib fracture identification, adrenal gland abnormalities, brain vessel classification, and synapse classification. This diversity ensures that our method has been evaluated on datasets with varying levels of complexity, resolution, and pathological variability, making it a robust benchmark for assessing the performance of machine learning methods. We have arranged our search space to balance prediction accuracy and model complexity for 3D tasks. The proposed NAS study is compared with state-of-the-art handcrafted networks, recent AutoML frameworks and NAS studies in terms of Area under the curve (AUC) and accuracy (ACC). The proposed NAS-based methods, particularly Ours #3†, demonstrate superior performance. It achieves the highest average AUC of 0.915 and ACC of 0.847 (best result across three independent runs), outperforming all handcrafted networks. Additionally, all proposed networks (Ours #1, #2, and #3) outperform handcrafted networks in average AUC and ACC, and they show superior performance for all individual datasets in terms of ACC. When compared to AutoML frameworks, our NAS-based methods consistently show better results. Ours #3 surpasses the best AutoML framework, auto-sklearn, which has an average AUC of 0.815 and ACC of 0.765. In comparisons with other NAS-based methods, all proposed methods † show superior performance in average AUC. They achieve the best ACC scores for 4 out of 6 datasets and the best AUC scores for 5 out of 6 datasets without performing data augmentation, unlike other studies. We have also performed an ablation study to evaluate cross-dataset performance of the proposed method with fine-tuning. We have selected the best-performing model from the NAS on the VesselMNIST3D dataset and trained it from scratch on five other datasets. The findings reveal a consistent decrease in both AUC and ACC across all datasets when compared to the best results achieved through dataset-specific NAS. The average differences are 0.019 for AUC and 0.028 for ACC, indicating a noticeable drop in performance. Despite this decline, the ablation results still outperform ResNets in terms of average AUC and ACC, showing superiority in 4 out of 5 datasets. Similarly, the ablation results are significantly better than those of AutoML frameworks in terms of average AUC and ACC, except for the Nodule3D dataset. In terms of computational complexity, our proposed methods significantly reduced complexity compared to ResNet models, with FLOPs reduced by up to 45.51 times and parameters by up to 211 times. Additionally, these methods outperform ResNet models in terms of AUC and ACC across most datasets, demonstrating both efficiency and superior performance. The main contributions of our study are as follows:

- We have extended the PBC-NAS method for 3D medical image classification tasks to ensure a balance between prediction accuracy and model complexity.
- We have performed an ablation study to evaluate the cross-dataset performance of the proposed methods with fine-tuning. We have demonstrated that we can achieve highly competitive results without individually performing NAS for each dataset.
- We have publicly shared the source code of this study and the best-performing model weights for each dataset.

Section 2 presents the detail of the proposed NAS method, search space and encoding steps. The Experimental Details (Sect. 3) provides information about the six different 3D datasets from MedMNIST3D and describes the training and evaluation process. The results and ablation studies are

provided and discussed in Sect. 4. Lastly, Sect. 5 presents the conclusion and future works.

2 Materials and methods

In this study, we explore the further enhancements of the PBC-NAS proposed in [28] for 3D medical image classification problems. The architectural structure used in PBC-NAS is derived from the cell-based network architecture introduced in [33, 34]. It is composed of multiple sequential stacks for feature extraction from the input image, followed by global average pooling and classifier. The stacks contain a number of modules, each of which has a cell-based network structure, denoted by a directed acyclic graph (DAG). The proposed architecture is illustrated in Fig. 1. The proposed method introduces several key modifications to the original PBC-NAS framework [28]. First, the operation set O is entirely replaced with 3D convolution operations to capture volumetric spatial relationships inherent in 3D medical images, whereas the original PBC-NAS employs 2D convolution operations designed for 2D peripheral blood cell images. Second, the search space is adjusted to account for the higher computational cost of 3D convolutions: the number of stacks (n_S) and modules (n_M) are reduced from $\{1, 2, 3\}$ to $\{1, 2\}$, and the number of initial feature maps (n_f^{init}) is reduced from $\{32, 64, 128\}$ to $\{8, 16, 32\}$. These adjustments ensure that the generated architectures remain computationally feasible while preserving sufficient representational capacity. Unlike the original PBC-NAS, which is evaluated on a single 2D dataset, the proposed method is evaluated on six diverse 3D datasets from MedMNIST3D [7], demonstrating its generalizability across different medical imaging modalities and classification tasks. Detailed information on the extended architecture from PBC-NAS, including the search space, the encoding, and the search algorithm, is presented in the following subsections.

2.1 Search space

The proposed method uses a DAG, consisting of a set of nodes (N) and a set of edges (E), to represent the cell structures within each module. In the graph, each node corresponds to a 3D convolution operation to be picked, and the edges depict the flow of information between these operations. Additionally, we consider some constraints to reduce the model complexity of the generated networks and avoid increasing the search space size: (1) the maximum number of nodes ($|N|$) is set to 7, comprising an input node, an output node, and 5 intermediate nodes. It should be noted that the input and output nodes are fixed. (2) the maximum number of edges ($|E|$) is set to 9. An edge exists between the input and output nodes, ensuring that input information is consistently relayed to the output layer to minimize information loss. The limits for the number of nodes ($|N| = 7$) and edges ($|E| = 9$) are adopted from the NAS-Bench-101 benchmark [33], which defines a widely used cell-based search space for reproducible NAS research. This configuration has been validated in prior NAS studies for biomedical image classification [41]. These studies demonstrate that it provides sufficient representational capacity while keeping the search space tractable. These constraints also prevent the generation of excessively complex architectures, which is particularly important in the 3D medical imaging setting due to the high computational cost of volumetric convolutions. (3)

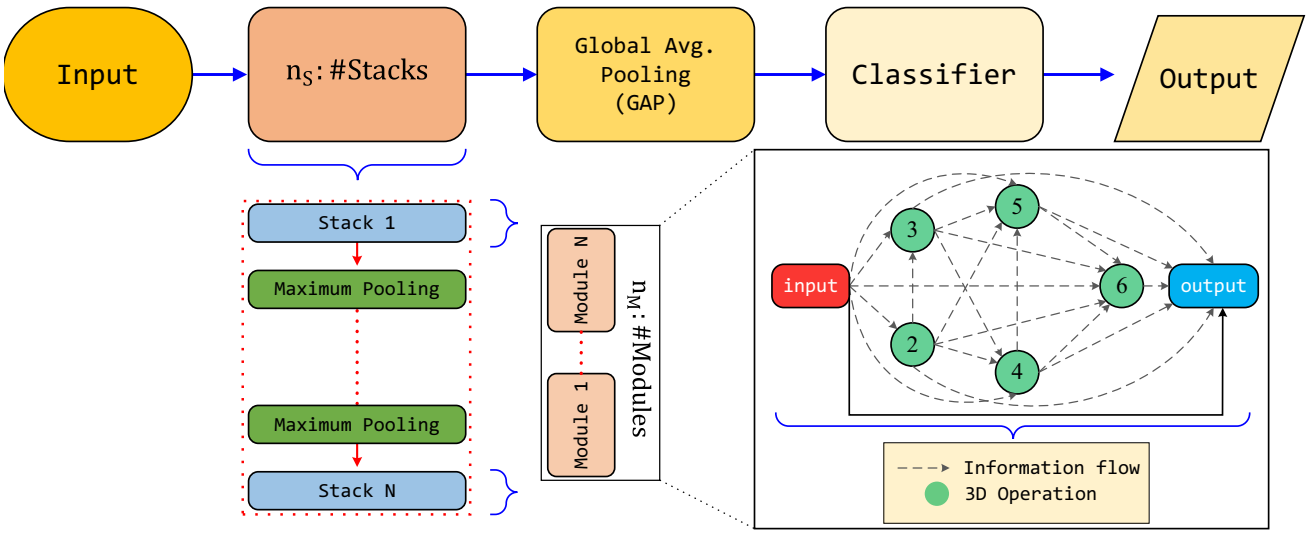

Fig. 1 Overview of the extended architecture from PBC-NAS. It comprises #Stacks that contain modules, and each module includes the structure of the cell. Each cell is used for feature extraction. Each stack is followed by a Global Average Pooling operation and classifier

Table 1 3D operations in the set of O. *IN* indicates the Instance normalization; *Mish* and *ReLU* indicate the Activation functions

#	3D operation
1	$3 \times 3 \times 3 \rightarrow IN \rightarrow ReLU$
2	$5 \times 5 \times 5 \rightarrow IN \rightarrow ReLU$
3	$7 \times 7 \times 7 \rightarrow IN \rightarrow ReLU$
4	$3 \times 3 \times 3 \rightarrow IN \rightarrow Mish$
5	$5 \times 5 \times 5 \rightarrow IN \rightarrow Mish$
6	$7 \times 7 \times 7 \rightarrow IN \rightarrow Mish$
7	$IN \rightarrow Mish \rightarrow 3 \times 3 \times 3$
8	$IN \rightarrow Mish \rightarrow 5 \times 5 \times 5$
9	$IN \rightarrow Mish \rightarrow 7 \times 7 \times 7$
10	$ReLU \rightarrow 3 \times 3 \times 3$
11	$ReLU \rightarrow 5 \times 5 \times 5$
12	$ReLU \rightarrow 7 \times 7 \times 7$
13	$Mish \rightarrow 3 \times 3 \times 3$
14	$Mish \rightarrow 5 \times 5 \times 5$
15	$Mish \rightarrow 7 \times 7 \times 7$

the number of 3D operations, which constitute the set O, is set to 15. Table 1 lists these 3D operations on the set O that are reported to be successful in [35]. We have selected and updated the 3D operations utilized in this study based on [35].

In our method, the number of stacks (n_S), the number of modules (n_M), and the number of initial feature maps (n_f^{init}) are defined as decision variables. These variables are incorporated into the search space for the dynamic network structure generation and effective complexity control. We have adjusted the search space size for these three decision variables to reduce model complexity: n_S and n_M are reduced from {1, 2, 3} to {1, 2}, and n_f^{init} from {32, 64, 128} to {8, 16, 32}.

2.2 Encoding

In the proposed method, a real-valued array of length 29 is used as a solution representation, encoding the components of a network architecture. Each entry in the array is in the range [0, 1]. Therefore, a mapping scheme is applied before evaluating the network architecture. The encoding process follows the methodology proposed by Awad et al. [36]. Fig. 2 demonstrates an example of the 5-step mapping scheme: (1) create the DAG; (2) pick the appropriate 3D operation for each node; (3) configure the value for the relevant parameters; (4) create the cell structures within each module; (5) build the modules for each stack, organize them within the stacks, and construct the complete architecture. The first 26 elements in the array are used to construct the cell structure, while the last 3 elements indicate the decision variables, n_S, n_M, and n_f^{init}. We use the first 21 elements in the array to determine the connection between nodes, represented by an upper triangular (0, 1)-matrix (0 means no edge between nodes, while 1 denotes an edge). The first element indicates if there is a connection between Node 1 and Node 2. If the value in the first element of the array is in [0, 0.5), an edge is placed between these two nodes. On the other hand, the remaining 5 elements show the 3D convolution operators ($o \in O$) associated with the nodes. The interval [0, 1] is split into 15 equal sub-intervals since there are 15 possible operations, each with a width of $1/15 \approx 0.067$. The right-open interval [0, 0.067) corresponds to the first operation given in the first row of Table 1. In other words, if the corresponding element falls within [0, 0.067), the first operation is selected for that node. The next right-open interval [0.067, 0.133) corresponds to the second operation,

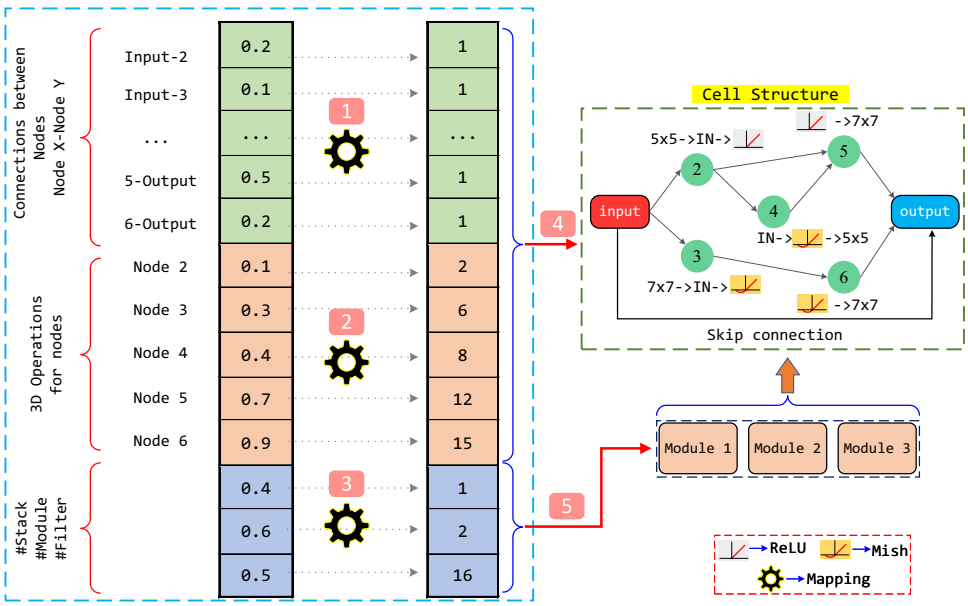

Fig. 2 Encoding process: (1) define the link between nodes; (2) choose 3D operations; (3) assign the values of the additional decision variables; (4) create the cell structures for each module; (5) assemble the modules within each stack, arrange them in the stacks, and construct the overall architecture

and this uniform partitioning continues for all 15 operations up to the interval [0.933, 1]. Similarly, the interval ([0, 1]) is divided into 2 sub-intervals for the decision variables, $n_S \in \{1, 2\}$, $n_M \in \{1, 2\}$, and is divided into 3 for the $n_f^{init} \in \{8, 16, 32\}$.

2.3 Search method

This study uses the Opposition-based Differential Evolution (ODE) algorithm as the search strategy. As described by Rahnamayan et al. [37], ODE is a refined version of the Differential Evolution (DE) algorithm. It incorporates opposition-based learning (OBL) into the steps of population initialization and generation jumping. OBL evaluates both candidate solutions and their opposites, offering additional insights into the search space, potentially enhancing exploration and accelerating convergence.

We have followed the ODE steps proposed in recent NAS studies [38]. This ODE involves these steps: **(1)** opposition-based initialization; **(2)** execution of DE operators, including mutation, crossover, and selection; **(3)** a probabilistic opposition-based generation jumping step. The search begins with a random population initialization (P) similar to traditional DE, followed by computing the opposites of these initial solutions (OP). Both candidate solutions and their opposites are assessed and combined. The best solutions from this combined group ($P \cup OP$) are chosen to form the initial population ($P \leftarrow best(P \cup OP)$), representing the opposition-based initialization in ODE. Following this, the operators from the standard DE variant(DE/rand/1/bin) [39] are performed, and new mutated solutions are generated and evaluated for the selection to the next population. Additionally, it includes a probabilistic opposition-based generation jumping step, where opposites of current candidate solutions (OP^*) are calculated and evaluated in the current population (P). The P and opposites of it (OP^*) are combined, and superior solutions are chosen to update the P.

Figure 3 presents the general steps of the ODE. The NAS process begins by randomly creating the initial population (P), and then calculating the opposite of each solution (OP). Subsequently, each generated candidate solution in $P \cup OP$ is evaluated by following the encoding steps explained in Sect. 2.2. The best NP solutions are then selected to form a parent population (P). Mutation and crossover operations are applied to the solutions in P to create mutated solutions, and a selection process is used to form an offspring population. Generation jumping steps are executed based on a probability condition ($Rnd \leq JR$). These steps are repeated until the termination condition, as described in Sect. 3, is met.

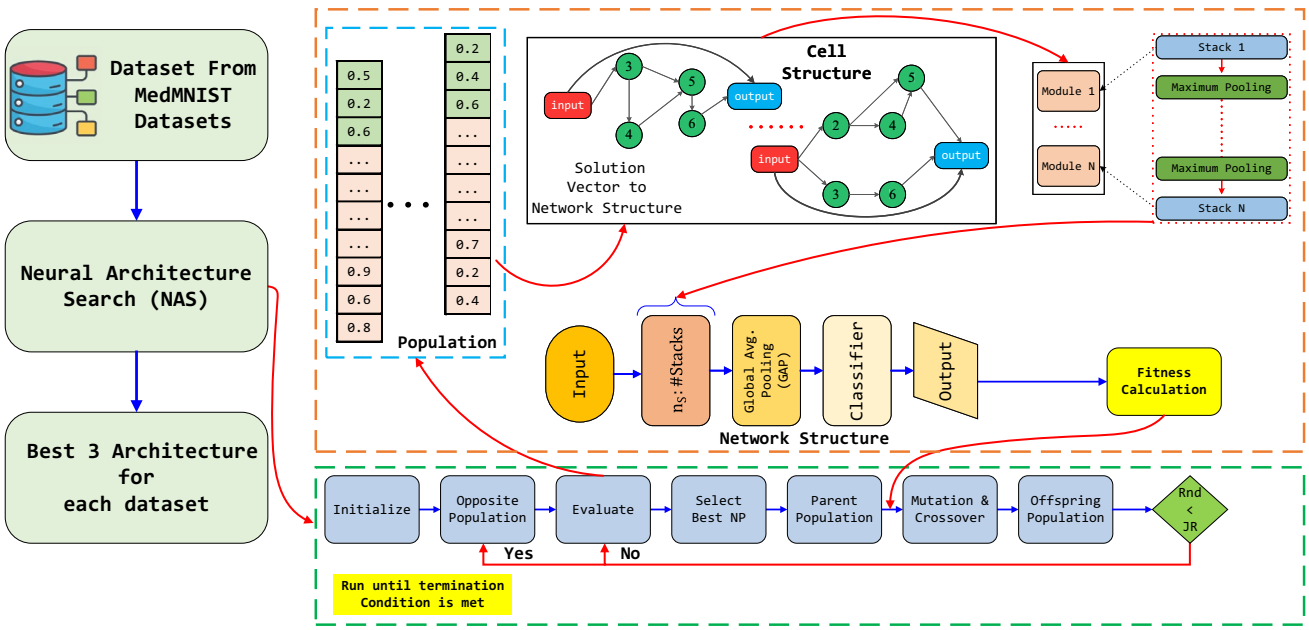

Fig. 3 The proposed NAS method steps from the dataset input to the selection of the best three architectures. Key components include population initialization based on opposition, evaluation of candidate network structures, and the standard DE operators: mutation, crossover, and selection. A probabilistic opposition-based generation jumping step is also incorporated, in which the opposites of current candidate solutions are computed and evaluated to enhance exploration of the search space. The process iterates until the termination condition is met, optimising cell structures and network configurations to maximise classification performance

3 Experimental details

3.1 Dataset

In this study, we have evaluated generated architectures on six different 3D datasets from MedMNIST3D [7]. OrganMNIST3D uses 3D bounding boxes derived from abdominal CT scans to classify 11 body organs. It contains 971 train, 161 validation, and 610 test images. NoduleMNIST3D consists of 3D CT scans, where each scan is labeled to indicate the presence or absence of lung nodules, which are small masses of tissue in the lungs that can be benign or malignant. The dataset comprises 1158 trains, 165 validation, and 310 test images. AdrenalMNIST3D includes images from CT scans labeled to assist in identifying and analyzing conditions related to the adrenal glands, such as tumors or other abnormalities. It comprises 1188 train, 98 validation, and 298 test images. FractureMNIST3D comprises CT scans from three different rib fractures and has 1027 training, 103 validation, and 240 test images. VesselMNIST3D provides 3D medical images for binary classification, typically derived from brain MRA. It includes 1335 train, 191 validation, and 382 test images. SynapseMNIST3D utilizes 3D image volumes of an adult rat using a multi-beam scanning electron microscope. It is specifically designed to classify synapses as either excitatory or inhibitory and comprises 1230 training, 177 validation, and 352 test images. We have used officially published train, validation, and test splits to evaluate generated models.

3.2 Experimental setup

The NAS process generates numerous candidate architectures, initially trained on the training set and evaluated on the validation set. The actual performance of these architectures is then evaluated on the test set. However, due to the large number of samples in the training and evaluation sets, it is computationally infeasible to train each candidate architecture on the entire dataset. Therefore, we randomly chose only a quarter of the training and validation sets for the training and evaluation of each candidate architecture during the NAS process. This approach follows the widely adopted low-fidelity or downscaled dataset evaluation strategy in the NAS literature [23, 40, 42–44], in which a reduced data subset is used to efficiently rank candidate architectures rather than to determine their final performance. The FABOLAS supports the claim that small subsets are quite representative for finding good parameters [44]. In order to improve the search process, we use early stopping techniques. Each candidate architecture is trained for a maximum of 50 epochs. However, if a candidate's validation loss does not improve for five consecutive epochs, the training is terminated at that epoch. The NAS process stops after evaluating 200 candidate architectures [38]. Following the NAS phase, the top three architectures are selected based on their validation accuracy. These architectures are then trained on the entire training set and evaluated on the test set, which is not included in the NAS phase, to report their actual performance. For both the training and evaluation phases, we use the following common hyper-parameters: the Adam optimizer with a learning rate of 1e-3, a batch size of 128, and BCELoss for binary classification tasks and CrossEntropyLoss for multi-class tasks. All experiments are conducted on the following setup: RTX 4090 GPU, 128GB of RAM, and Intel i9 processor. We have not performed any pre-processing, data augmentation, or feature selection techniques. We have analyzed the computational cost of the NAS process and the resource requirements of the resulting models to demonstrate the feasibility of our approach in resource-constrained environments. The reported NAS training times reflect the full search cost, encompassing the evaluation of all 200 candidate architectures. Each architecture is trained for a maximum of 50 epochs on one quarter of the training and validation sets, with early stopping applied. The required NAS search times for the six MedMNIST3D datasets (AdrenalMNIST3D, FractureMNIST3D, NoduleMNIST3D, OrganMNIST3D, SynapseMNIST3D, and VesselMNIST3D) are 88, 60, 105, 115, 63, and 124 min, respectively, on an NVIDIA RTX 4090 GPU with a maximum GPU memory usage of 4GB. Following the NAS phase, the top three selected architectures are trained on the full training set for each dataset. The training time for these final architectures ranges from approximately 30 to 120 min per run, depending on the dataset size and the architectural complexity of the selected model. The source code and best-performing model weights are shared on Github.

4 Results

We compare the performances of various methods on six different 3D medical image classification datasets. The methods are categorized into handcrafted networks, AutoML frameworks, and NAS studies. Table 2 shows the results for each dataset and also the average AUC and ACC scores. The reported average results from MedMNIST are used for ResNet-18, ResNet-50, auto-sklearn, and Auto-Keras. Other studies ([10, 26, 27]) have not explicitly stated whether the reported results are average or best. Therefore, we have reported the best and average results from three different runs for a fair comparison. Additionally, some studies perform data augmentation and neural architecture search, which we indicated with the abbreviations DA and NAS.

Table 2 The detailed comparison of the handcrafted networks, AutoML frameworks, and NAS studies on MedMNIST3D datasets

Methods	DA	NAS	Organ3D		Nodule3D		Fracture3D		Adrenal3D		Vessel3D		Synapse3D		Average	
			AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
ResNet-18 + 2.5D	✗	✗	0.977	0.788	0.838	0.835	0.587	0.451	0.718	0.772	0.748	0.846	0.634	0.696	0.750	0.731
ResNet-18 + 3D	✗	✗	*0.996*	0.907	0.863	0.844	0.712	0.508	0.827	0.721	0.874	0.877	0.82	0.745	0.849	0.767
ResNet-18 + ACS	✗	✗	0.994	0.9	0.873	0.847	0.714	0.497	0.839	0.754	0.93	0.928	0.705	0.722	0.843	0.775
ResNet-50 + 2.5D	✗	✗	0.974	0.769	0.835	0.848	0.552	0.397	0.732	0.763	0.751	0.877	0.669	0.735	0.752	0.732
ResNet-50 + 3D	✗	✗	0.994	0.883	0.875	0.847	0.725	0.494	0.828	0.745	0.907	0.918	0.851	0.795	0.863	0.780
ResNet-50 + ACS	✗	✗	0.994	0.889	0.886	0.841	*0.75*	0.517	0.828	0.758	0.912	0.858	0.719	0.709	0.848	0.762
Shin et al. [10]	✗	✗	*0.996*	0.939	0.868	0.845	0.595	0.421	0.768	0.516	0.453	0.887	0.54	0.73	0.703	0.723
FPViT [17, 18]	✗	✗	0.923	0.800	0.814	0.822	0.640	0.438	0.801	0.704	0.770	0.888	0.530	0.712	0.746	0.727
ViT-3D [20]	✗	✗	–	–	***0.923***	*0.896*	0.651	0.545	0.838	0.828	0.837	0.911	–	–	–	–
auto-sklearn	✗	✓	0.977	0.814	*0.914*	0.874	0.628	0.453	0.828	0.802	0.91	0.915	0.631	0.73	0.815	0.765
AutoKeras	✗	✓	0.979	0.804	0.844	0.834	0.642	0.458	0.804	0.705	0.773	0.894	0.538	0.724	0.763	0.737
TPOT [10]	✗	✓	0.977	0.789	0.912	0.877	0.619	0.458	0.765	0.768	0.796	0.893	0.601	0.73	0.778	0.753
AutoGluon [10]	✗	✓	0.942	0.649	0.631	0.793	0.525	0.376	0.768	0.516	0.453	0.887	0.54	0.73	0.643	0.659
Ali et al. [27]	✓	✓	0.995	0.908	0.871	0.877	0.728	***0.690***	0.857	0.805	0.940	0.940	0.820	0.846	0.869	*0.844*
MedPipe [26]	✓	✓	0.964	***0.963***	0.913	***0.918***	0.575	0.579	0.826	0.808	0.950	0.950	0.801	0.812	0.838	0.838
Ours #1*	✗	✓	0.995	0.911	0.913	0.873	0.707	0.557	0.891	0.84	0.972	0.961	0.782	0.782	0.877	0.821
Ours #2*	✗	✓	0.994	0.912	0.894	0.855	0.733	0.585	0.877	0.825	0.973	0.955	0.887	0.851	0.893	0.831
Ours #3*	✗	✓	0.998	0.938	0.913	0.865	0.764	0.593	0.888	0.834	0.983	0.963	0.888	0.835	0.906	0.838
Ours #1†	✗	✓	0.995	0.923	0.911	0.890	0.730	0.579	*0.884*	***0.842***	0.954	***0.963***	0.839	0.796	0.886	0.832
Ours #2†	✗	✓	0.992	0.917	0.873	0.865	0.743	0.600	0.879	0.832	*0.968*	*0.958*	*0.897*	***0.872***	*0.892*	0.841
Ours #3†	✗	✓	***0.998***	*0.944*	0.911	0.874	***0.768***	*0.608*	***0.892***	*0.836*	***0.987***	***0.963***	***0.934***	*0.858*	***0.915***	***0.847***

The Bold Italic and Italic colors indicate the best and second-best results, respectively

The average and best results for the proposed methods are reported

DA Data augmentation, *NAS* Neural architecture search

*: average results obtained from three different runs

†: best results-: result is not reported

In Table 2, #Ours 1, 2, and 3 indicate the top three models obtained from the NAS process.

The handcrafted networks, which include variations of ResNet-18 and ResNet-50, generally show competitive performance across the datasets. For instance, ResNet-18 + 3D achieves a high AUC of 0.996 on the Organ3D dataset, among the best results for handcrafted networks. However, the average performance across all datasets for these networks tends to be lower than the NAS-based methods. The average AUC and ACC for the best-performing handcrafted network, ResNet-50 + 3D, are 0.863 and 0.780, respectively. In contrast, our NAS-based methods demonstrate superior performance, especially Ours #3. Ours #3† achieves the highest average AUC of 0.915 and ACC of 0.847 (best result across three independent runs), outperforming all handcrafted networks. The average-run result, Ours #3*, achieves an average AUC of 0.906 and ACC of 0.838, which also surpasses all handcrafted networks. Additionally, all proposed networks (Ours #1, #2, and #3) outperform handcrafted networks in average AUC and ACC, and they show superior performance for all individual datasets in terms of ACC. We have conducted a comparison of our method with vision-transformer-based architectures [17, 20]. Our method demonstrates significant performance improvements over FPViT across all six MedMNIST3D datasets. The best-performing model (Ours #3†) surpasses FPViT by achieving an average AUC that is 16.9% higher (0.915 compared to 0.746) and an average accuracy that is 12% higher (0.847 compared to 0.727). Significant improvements are observed on Organ3D (0.998 versus 0.923 AUC) and Synapse3D (0.934 versus 0.530 AUC). Regarding ViT-3D [20], results are available for only four of the six datasets, namely NoduleMNIST3D, FractureMNIST3D, AdrenalMNIST3D, and VesselMNIST3D, as the original authors did not report results for OrganMNIST3D and SynapseMNIST3D. Therefore, the comparison with ViT-3D is limited to these four datasets. Among these, the proposed method outperforms ViT-3D on FractureMNIST3D (0.768 versus 0.651 AUC), AdrenalMNIST3D (0.892 versus 0.838 AUC), and VesselMNIST3D (0.987 versus 0.837 AUC). On NoduleMNIST3D, ViT-3D achieves a slightly higher AUC (0.923 versus 0.911); however, the proposed method remains competitive on this dataset. No comparison with ViT-3D is made for OrganMNIST3D and SynapseMNIST3D, as results for these datasets are not reported in the original publication [20].

In summary, the results demonstrate the effectiveness of our NAS-based approaches over traditional handcrafted networks.

The comparison between AutoML frameworks and our NAS-based methods reveals a clear performance advantage for our approaches across average scores and individual datasets. In terms of average performance, our methods consistently outperform the AutoML frameworks. Ours #3 achieves the highest average AUC of 0.906 and ACC of 0.838, surpassing the best AutoML framework, autosklearn, which gives an average AUC of 0.815 and ACC of 0.765. Similarly, Ours #1 and Ours #2 also outperform the AutoML frameworks, with average AUCs of 0.877 and 0.893 and ACCs of 0.821 and 0.831, respectively. Furthermore, our methods consistently outperform AutoML frameworks across all datasets. For the Organ3D, Ours #3 achieves an AUC of 0.998 and ACC of 0.938, significantly better than AutoML's best of 0.979 AUC and 0.814 ACC. In the Fracture3D and Adrenal3D datasets, all proposed models demonstrate superior AUC and ACC scores compared to the AutoML frameworks. The Vessel3D and Synapse3D show a similar trend; all proposed methods give significantly better performance than AutoML frameworks. These notable differences highlight the effectiveness of our NAS-based methods in optimizing network architectures for complex 3D medical image classification tasks. In conclusion, our NAS-based methods, particularly Ours #3, consistently outperform the AutoML frameworks across all datasets and metrics.

The comparative analysis of the methods Ali et al. [27], MedPipe [26], and the proposed methods on the MedMNIST3D datasets reveals distinct performance characteristics across various metrics. It is important to note that among the compared methods, Ali et al. [27] and MedPipe [26] employ data augmentation techniques during training. In both cases, augmentation is applied solely to the training data, and the test dataset remains unchanged. All methods, including the proposed approach, are evaluated on the same publicly shared test split from MedMNIST [7]. Therefore, the comparison is conducted under consistent evaluation conditions. To ensure transparency, methods that utilize data augmentation are explicitly marked with the abbreviation *DA* in Table 2. Ali et al. demonstrate strong performance by employing both data augmentation and neural architecture search, achieving an average AUC of 0.869 and an ACC of 0.844. They perform well in the Fracture3D dataset, achieving the highest ACC of 0.690 among the compared methods. MedPipe also uses data augmentation and neural architecture search, achieving a remarkable ACC of 0.963 on the Organ3D dataset, marking the highest accuracy in that dataset. Additionally, MedPipe shows competitive results in the Nodule3D dataset, with an AUC of 0.913 and an ACC of 0.918. In contrast, the proposed methods, Ours #1, #2, and #3, do not employ data augmentation but perform neural architecture search. Despite this, they demonstrate superior performance across several metrics.

Ours #3† achieves the best average AUC and ACC scores, and it gives the highest AUC scores for Fracture3D,

Adrenal3D, Vessel3D, and Synapse3D. Furthermore, Ours #3† gives better ACC scores than other methods in Vessel3D and achieves second-bests ACC scores for the Organ3D, Fracture3D, Adrenal3D, and Synapse3D datasets. Overall, all proposed methods† show superior performance in average AUC. They achieve the best ACC scores for 4 out of 6 datasets and the best AUC scores for 5 out of 6 datasets. The single exception in AUC is the Nodule3D dataset, where ViT-3D [20] achieves a marginally higher AUC of 0.923 compared to 0.911 for Ours #3†. This difference may be attributed to the use of a pre-trained transformer-based architecture in ViT-3D, which benefits from large-scale pre-training not employed in the proposed method. Regarding ACC, the proposed method does not achieve the best result on Organ3D, Nodule3D, and Fracture3D. On Organ3D and Nodule3D, MedPipe [26] achieves higher ACC scores of 0.963 and 0.918, respectively, compared to 0.944 and 0.874 for Ours #3†. On Fracture3D, Ali et al. [27] achieve the highest ACC of 0.690, compared to 0.608 for Ours #3†. Notably, both MedPipe and Ali et al. employ data augmentation during training, which provides an additional performance advantage on these datasets. Additionally, all proposed methods⋆ provide the best average AUC scores compared to others and achieve highly competitive results for the average ACC and individual datasets. These achievements highlight the effectiveness of the proposed NAS approach, demonstrating that even without data augmentation, the methods can achieve state-of-the-art performance. This highlights the potential of NAS to optimize neural network architectures effectively, achieving high performance across diverse datasets.

Additionally, all proposed methods⋆ provide the best average AUC scores compared to others and achieve highly competitive results for the average ACC and individual datasets. These results highlight the effectiveness of the proposed NAS approach in optimising neural network architectures for 3D medical image classification without relying on data augmentation.

We have also conducted a Wilcoxon signed-rank test ($p = 0.05$) to validate the statistical significance of the performance improvements achieved by our proposed models (see Table 3). This test uses the ACC and AUC values obtained from three independent runs of the best-reported models across six datasets (18 data points per comparison - Model $\times$ three runs $\times$ six datasets). The analysis included comparisons with ResNet18, ResNet50, Auto-sklearn, and AutoKeras, as these methods provide publicly available results for multiple runs. The results demonstrate that our models (Ours #1⋆, Ours #2⋆, and Ours #3⋆) significantly outperform the baseline methods, with p-values consistently below 0.05 for both ACC and AUC metrics. For instance, Ours #3⋆ achieves $p = 1.52 \times 10^{-5}$. for ACC and $p = 7.62 \times 10^{-6}$. for AUC when compared to ResNet18 (2.5D), and similarly significant results against AutoML frameworks such as AutoKeras. These findings provide strong statistical evidence for the robustness of our NAS-based models.

4.1 Cross-dataset performance with fine-tuning

The ablation study evaluates the cross-dataset performance of a model optimized for the VesselMNIST3D through NAS. VesselMNIST3D is selected as the source dataset because it contains the largest number of training samples (1,335) among the six MedMNIST3D datasets. However, it should be noted that the margin over the next largest datasets, SynapseMNIST3D (1,230) and AdrenalMNIST3D (1,188), is relatively small. Therefore, the choice of VesselMNIST3D as the source dataset is to some extent arbitrary, and the cross-dataset fine-tuning results may vary if a different source dataset is selected. This constitutes a limitation of the ablation study, and future work may consider repeating the experiment with alternative source datasets to assess the sensitivity of the findings to this choice. The main objective is determining whether this model can achieve similar results when fine-tuned on other datasets without dataset-specific NAS. We select the best-performing model as a result of NAS on the VesselMNIST3D dataset and then train it from scratch on five other datasets: Organ3D, Nodule3D, Fracture3D, Adrenal3D, and Synapse3D. The results are compared with the best results obtained through NAS

Table 3 The p-values from the Wilcoxon signed-rank test ($p = 0.05$) comparing the ACC and AUC values of the proposed models (Ours #1⋆, Ours #2⋆, and Ours #3⋆) with baseline methods (ResNet18, ResNet50, Auto-sklearn, and AutoKeras) across six datasets and three independent runs

	ResNet-18 + 2.5D	ResNet-18 + 3D	ResNet-18 + ACS	ResNet-50 + 2.5D	ResNet-50 + 3D	ResNet-50 + ACS	AutoKeras	auto-sklearn
ACC								
Ours #1⋆	7.62e−06	1.90e−04	3.28e−04	1.52e−05	2.28e−05	1.06e−04	7.62e−06	7.62e−05
Ours #2⋆	7.62e−06	6.71e−04	2.33e−04	7.62e−06	1.38e−02	3.81e−05	7.62e−06	3.81e−05
Ours #3⋆	1.52e−05	3.81e−05	1.52e−05	1.52e−05	5.34e−05	2.28e−05	1.52e−05	1.06e−04
AUC								
Ours #1⋆	7.62e−06	1.57e−04	2.33e−04	7.62e−06	8.96e−04	5.59e−04	7.62e−06	5.34e−05
Ours #2⋆	7.62e−06	2.68e−04	5.34e−04	7.62e−06	1.65e−02	1.20e−02	7.62e−06	5.34e−05
Ours #3⋆	7.62e−06	7.62e−06	7.62e−06	7.62e−06	4.19e−04	1.57e−03	7.62e−06	3.28e−04

Values below 0.05 indicate statistically significant differences

Table 4 The results of the cross-dataset performance with fine-tuning. The difference indicates the difference between the best result and the ablation result

Methods	Comparison with Ours*											
	Organ3D		Nodule3D		Fracture3D		Adrenal3D		Synapse3D		Average	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
Ablation Result	0.990	0.892	0.893	0.855	0.736	0.564	0.873	0.822	0.865	0.820	0.871	0.791
Best results	0.998	0.938	0.913	0.873	0.764	0.593	0.891	0.840	0.888	0.851	0.891	0.819
Difference	***0.008***	***0.046***	***0.020***	***0.018***	***0.028***	***0.029***	***0.018***	***0.018***	***0.023***	***0.031***	***0.019***	***0.028***
Methods	Comparison with Ours†											
	Organ3D		Nodule3D		Fracture3D		Adrenal3D		Synapse3D		Average	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
Ablation result	0.994	0.912	0.908	0.868	0.768	0.583	0.879	0.826	0.893	0.830	0.888	0.804
Best results	0.998	0.944	0.911	0.890	0.768	0.608	0.892	0.842	0.934	0.872	0.901	0.831
Difference	***0.004***	***0.032***	***0.003***	***0.022***	0.0	***0.025***	***0.013***	***0.016***	***0.041***	***0.042***	***0.012***	***0.027***

The Bold Italic represents the decreasing performance as a result of ablation

Comparison with Ours* covers Ours #1*, 2*, 3* results in Table 2

Comparison with Ours† covers the Ours #1†, 2†, 3† results in Table 2

for each dataset, with differences highlighted in red to indicate performance degradation due to the ablation. Table 4 provides a detailed comparison of the ablation results with the best results for each dataset (see Table 2). In the comparison with Ours*, the ablation results show a consistent decrease in both AUC and ACC across all datasets, with average differences of 0.019 for AUC and 0.028 for ACC. This indicates that while the model trained on VesselMNIST3D performs well when fine-tuned on other datasets, its performance drops noticeably compared to the best results obtained through dataset-specific NAS. Similarly, in the comparison with Ours†, the ablation results show a decrease in performance, with the differences being slightly smaller. The average differences are 0.012 for AUC and 0.027 for ACC. The Fracture3D dataset shows no difference in AUC, suggesting that the model's architecture may be well-suited to this particular dataset, even without dataset-specific NAS. Furthermore, the ablation results outperform ResNets in terms of average AUC and ACC, and are also superior in 4 out of 5 datasets in terms of AUC and ACC (see Table 2). Similar results are obtained for AutoML frameworks; ablation results are significantly better than AutoML frameworks in terms of average AUC and ACC. In addition, ablation results are better compared to AutoML frameworks in terms of AUC and ACC in all datasets (except Nodule3D). Overall, the ablation study demonstrates that while fine-tuning a model optimized for one dataset can achieve competitive results on other datasets, there is a trade-off in performance. The differences highlighted in red underscore the importance of dataset-specific NAS. These findings suggest that fine-tuning is a useful strategy for cross-dataset experiments, especially when resources are limited, but it may not fully substitute the benefits of conducting NAS for each dataset individually.

4.2 Comparison of model complexity

This section presents a detailed computational complexity comparison of various network models in terms of the number of trainable parameters (#Params) and floating point operations per second (FLOPs). We choose reported models in MedMNIST [7], and calculate #Params and FLOPs for these models. The primary focus of this analysis is to identify models with lower #Params and FLOPs, which indicate more efficient and potentially more effective models.

Table 5 shows a detailed comparison of ResNets models and proposed methods (Ours #) in terms of #Params and FLOPs. The ResNet models, particularly ResNet-18 and ResNet-50, show a consistent pattern in their parameter and FLOP counts across all tasks. For example, ResNet-18 + 2.5D and ResNet-18 + ACS have 11.168 million parameters and 5.94 giga FLOPs, while ResNet-18 + 3D significantly

Table 5 Comparison of model complexity across various 3D medical imaging tasks

Methods	Organ3D		Nodule3D		Fracture3D		Adrenal3D		Vessel3D		Synapse3D	
	#Params	FLOPs	#Params	FLOPs	#Params	FLOPs	#Params	FLOPs	#Params	FLOPs	#Params	FLOPs
ResNet-18 + 2.5D	11.168 M	5.94 G	11.168 M	5.94 G	11.168 M	**5.94 G**	11.168 M	5.94 G	11.168 M	5.94 G	11.168 M	5.94 G
ResNet-18 + 3D	33.140 M	17.75G	33.140 M	17.75G	33.140 M	17.75G	33.140 M	17.75G	33.140 M	17.75G	33.140 M	17.75G
ResNet-18 + ACS	11.168 M	5.94 G	11.168 M	5.94 G	11.168 M	**5.94 G**	11.168 M	5.94 G	11.168 M	5.94 G	11.168 M	5.94 G
ResNet-50 + 2.5D	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G
ResNet-50 + 3D	46.138 M	23.94 G	46.138 M	23.94 G	46.138 M	23.94 G	46.138 M	23.94 G	46.138 M	23.94 G	46.138 M	23.94 G
ResNet-50 + ACS	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G	23.503 M	12.49 G
Ours #1	**0.157 M**	**3.44 G**	0.317 M	2.09 G	2.349 M	15.81 G	**0.018 M**	**0.39 G**	**0.237 M**	**1.57 G**	1.719 M	37.70 G
Ours #2	0.240 M	5.27 G	**0.255 M**	**1.68 G**	**0.731 M**	16.05 G	3.040 M	20.02 G	0.509 M	3.36 G	**0.218 M**	**4.78 G**
Ours #3	0.859 M	18.86 G	0.315 M	2.05 G	1.200 M	7.91 G	1.283 M	8.46 G	1.200 M	7.91 G	1.016 M	22.28 G

The table lists the number of trainable parameters (#Params) and floating point operations per second (FLOPs) for different network architectures

The bold values indicate the best results for the corresponding measure

#Params are reported in megabyte, and FLOPs are reported in gigabyte

increases these metrics to 33.140 million and 17.75 giga FLOPs. Similarly, ResNet-50 variants significantly increase complexity, with the 3D version reaching 46.138 million parameters and 23.94 giga FLOPs. In contrast, our methods ("Ours") demonstrate a remarkable reduction in both #Params and FLOPs, suggesting a more efficient design. For example, Ours #1 achieves the lowest parameter count of 0.157 million and 3.44 giga FLOPs for the Organ3D task and even lower metrics for Adrenal3D with 0.018 million parameters and 0.39 giga FLOPs. Ours #2 and Ours #3 also show competitive performance, with Ours #2 achieving the lowest FLOPs for Nodule3D at 1.68 giga. For the Organ3D dataset, Ours #1 achieves a remarkable reduction in computational complexity, with FLOPs approximately 1.73 times lower than the ResNet-18 + 2.5D and ACS models and about 5.16 times lower than the ResNet-18 + 3D model. This efficiency is followed in the Nodule3D dataset, where Ours #2 demonstrates a reduction in FLOPs by approximately 3.54 times compared to the ResNet-18 + 2.5D and ACS models and 10.57 times compared to the ResNet-18 + 3D model. In the Fracture3D dataset, Ours #2 again shows a significant advantage, with FLOPs about 1.49 times lower than the ResNet-18 + 2.5D and ACS models and 1.49 times lower than the ResNet-18 + 3D model. For the Adrenal3D dataset, Ours #1 achieves an impressive reduction, with FLOPs approximately 15.23 times lower than the ResNet-18 + 2.5D and ACS models and 45.51 times lower than the ResNet-18 + 3D model. The same patterns are observed for #Params; our methods consistently demonstrate a significant reduction compared to ResNet models across all datasets. Ours # reduce the number of trainable parameters by up to 211 times. Furthermore, in addition to achieving significant reductions in computational complexity, the "Ours" methods also demonstrate superior performance compared to traditional ResNet models in terms of AUC and ACC across most datasets (see Table 2). This dual advantage of reduced complexity and enhanced performance underscores the effectiveness of the proposed methods.

5 Conclusion

In this study, we have successfully extended the PBC-NAS method for 3D medical image classification tasks, achieving a balance between prediction accuracy and model complexity. Our NAS-based method, specifically Ours #3, has shown superior performance compared to state-of-the-art handcrafted networks, recent AutoML frameworks, and other NAS studies in terms of Area under the curve (AUC) and accuracy (ACC). The proposed method has achieved the highest average AUC of 0.915 and ACC of 0.847, outperforming all handcrafted networks and AutoML frameworks. Compared to

other NAS-based methods, all the proposed methods† exhibit superior performance in terms of average AUC, achieving the highest AUC scores in 5 out of 6 datasets and the best ACC scores in 4 out of 6 datasets. It should be noted that the competing NAS-based methods, namely Ali et al. [27] and MedPipe [26], employ data augmentation as an additional component of their pipelines. In contrast, the proposed methods achieve these results without utilizing any data augmentation, preprocessing, or feature selection, which underscores the effectiveness of the proposed NAS approach in optimizing neural network architectures under more constrained conditions. Additionally, our methods have shown significant reductions in computational complexity, with FLOPs reduced by up to 45.51 times and parameters by up to 211 times compared to ResNet models while still maintaining superior performance. The ablation study has revealed that while fine-tuning a model optimized for one dataset can achieve competitive results on other datasets, there is a noticeable drop in performance compared to dataset-specific NAS. This highlights the importance of performing NAS for each dataset individually to achieve optimal results. Despite this, the ablation results still outperformed ResNets and AutoML frameworks in terms of average AUC and ACC. Overall, our study demonstrates the effectiveness of the proposed NAS approach in optimizing neural network architectures for complex 3D medical image classification tasks, achieving state-of-the-art performance without the need for data augmentation. The source code and best-performing model weights for each dataset have been publicly shared, contributing to the advancement of research in this field. Future work will focus on further improving the search process and exploring additional datasets to validate the generalizability of the proposed methods.

Acknowledgements A. Kiraz acknowledges partial support from the Turkish Academy of Sciences (TüBA).

Author contributions Zeki Kuş: Conceptualization, Methodology, Software, Validation, Writing - Original Draft, Visualization. Berna Kiraz: Conceptualization, Methodology, Writing - Original Draft. Musa Aydin: Conceptualization, Methodology, Writing - Original Draft, Visualization. Alper Kiraz: Conceptualization, Methodology, Writing - Original Draft, Supervision.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Ethical approval and consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Wang, H., Zhang, Q., Lu, H., Won, D., Yoon, S.W.: 3d medical image classification with depthwise separable networks. Procedia. Manuf. **39**, 349–356 (2019). https://doi.org/10.1016/j.promfg.2020.01.369
2. Zhou, L., Liu, H., Bae, J., He, J., Samaras, D., Prasanna, P.: Self pre-training with masked autoencoders for medical image classification and segmentation. In: 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI), pp. 1–6 (2023). https://doi.org/10.1109/ISBI53787.2023.10230477
3. Abdelhamid, A., El-Ghamry, A., Abdelhay, E.H., Abo-Zahhad, M.M., Moustafa, H.E.: Improved pulmonary embolism detection in CT pulmonary angiogram scans with hybrid vision transformers and deep learning techniques. Sci. Rep. (2025). https://doi.org/10.1038/s41598-025-16238-4
4. Al-Hejri, A.M., Al-Tam, R.M., Fazea, M., Sable, A.H., Lee, S., Al-antari, M.A.: ETECADx: ensemble self-attention transformer encoder for breast cancer diagnosis using full-field digital X-ray breast images. Diagnostics **13**(1), 89 (2022). https://doi.org/10.3390/diagnostics13010089
5. Singh, S.P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., Gulyás, B.: 3d deep learning on medical images: a review. Sensors **20**(18), 5097 (2020). https://doi.org/10.3390/s20185097
6. Ilesanmi, A.E., Ilesanmi, T.O., Ajayi, B.O.: Reviewing 3d convolutional neural network approaches for medical image segmentation. Heliyon **10**(6), 27398 (2024). https://doi.org/10.1016/j.heliyon.2024.e27398
7. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Sci. Data (2023). https://doi.org/10.1038/s41597-022-01721-8
8. Gao, S., Li, X., Li, X., Li, Z., Deng, Y.: Transformer based tooth classification from cone-beam computed tomography for dental charting. Comput. Biol. Med. **148**, 105880 (2022). https://doi.org/10.1016/j.compbiomed.2022.105880
9. Manzari, O.N., Ahmadabadi, H., Kashiani, H., Shokouhi, S.B., Ayatollahi, A.: Medvit: A robust vision transformer for generalized medical image classification. Comput. Biol. Med. **157**, 106791 (2023). https://doi.org/10.1016/j.compbiomed.2023.106791
10. Shin, S., Park, D., Ji, S., Joo, G., Im, H.: Medical data analysis using automl frameworks. J. Electr. Eng. Technol. (2024). https://doi.org/10.1007/s42835-024-01919-3
11. Kiechle, J., Lang, D.M., Fischer, S.M., Felsner, L., Peeken, J.C., Schnabel, J.A.: Graph neural networks: a suitable alternative to MLPs in latent 3D medical image classification? (2024). arXiv:https://arxiv.org/abs/2407.17219
12. Dhiravidachelvi, E., Devadas, T.J., Kumar, P.J.S., Pandi, S.S.: Enhancing image classification using adaptive convolutional

autoencoder-based snow avalanches algorithm. SIViP **18**(10), 6867–6879 (2024). https://doi.org/10.1007/s11760-024-03357-0
13. Elbedwehy, S., Hassan, E., Saber, A., Elmonier, R.: Integrating neural networks with advanced optimization techniques for accurate kidney disease diagnosis. Sci. Rep. (2024). https://doi.org/10.1038/s41598-024-71410-6
14. Saber, A., Elbedwehy, S., Awad, W.A., Hassan, E.: An optimized ensemble model based on meta-heuristic algorithms for effective detection and classification of breast tumors. Neural Comput. Appl. **37**(6), 4881–4894 (2024). https://doi.org/10.1007/s00521-024-10719-9
15. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28. Curran Associates, Inc., New York (2015). https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf
16. Jin, H., Chollet, F., Song, Q., Hu, X.: Autokeras: an automl library for deep learning. J. Mach. Learn. Res. **24**(6), 1–6 (2023)
17. Liu, J., Li, Y., Cao, G., Liu, Y., Cao, W.: Feature pyramid vision transformer for medmnist classification decathlon. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2022). https://doi.org/10.1109/IJCNN55064.2022.9892282
18. Zheng, Z., Jia, X.: Complex mixer for MedMNIST classification decathlon. arXiv (2023). https://doi.org/10.48550/ARXIV.2304.10054
19. Schäfer, R., Nicke, T., Höfener, H., Lange, A., Merhof, D., Feuerhake, F., Schulz, V., Lotz, J., Kiessling, F.: Overcoming data scarcity in biomedical imaging with a foundational multi-task model. Nat. Comput. Sci. **4**(7), 495–509 (2024). https://doi.org/10.1038/s43588-024-00662-z
20. Lai, Z., Wu, J., Chen, S., Zhou, Y., Hovakimyan, N.: Residual-based language models are free boosters for biomedical imaging tasks. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 5086–5096 (2024). https://doi.org/10.1109/CVPRW63382.2024.00515
21. Olson, R.S., Bartley, N., Urbanowicz, R.J., Moore, J.H.: Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. GECCO '16. ACM, ??? (2016). https://doi.org/10.1145/2908812.2908918
22. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A.: AutoGluon-tabular: robust and accurate AutoML for structured data (2020). arXiv:https://arxiv.org/abs/2003.06505
23. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: a survey. J. Mach. Learn. Res. **20**(55), 1–21 (2019)
24. Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., Wang, X.: A comprehensive survey of neural architecture search: challenges and solutions. ACM Comput. Surv. **54**(4), 1–34 (2021). https://doi.org/10.1145/3447582
25. Kang, J.-S., Kang, J., Kim, J.-J., Jeon, K.-W., Chung, H.-J., Park, B.-H.: Neural architecture search survey: a computer vision perspective. Sensors **23**(3), 1713 (2023). https://doi.org/10.3390/s23031713
26. He, X., Chu, X.: Medpipe: end-to-end joint search of data augmentation and neural architecture for 3d medical image classification. In: 2023 IEEE International Conference on Medical Artificial Intelligence (MedAI), pp. 344–354 (2023). https://doi.org/10.1109/MedAI59581.2023.00053
27. Ali, M.J., Moalic, L., Essaid, M., Idoumghar, L.: Evolutionary Neural Architecture Search for 2D and 3D Medical Image Classification, pp. 131–146. Springer, Berlin (2024). https://doi.org/10.1007/978-3-031-63751-3_9
28. Kuş, Z., Kiraz, B., Aydın, M., Kiraz, A.: PBC-NAS: Neural architecture search for peripheral blood cells classification. In: 2024 32nd Signal Processing and Communications Applications Conference (SIU). IEEE, New York (2024).https://doi.org/10.1109/siu61531.2024.10601013
29. Bilic, P., Christ, P., Li, H.B., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G.E.H., Chartrand, G.: The liver tumor segmentation benchmark (lits). Med. Image Anal. **84**, 102680 (2023)
30. Armato, S.G.: The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. Med. Phys. **38**(2), 915–931 (2011). https://doi.org/10.1118/1.3528204
31. Jin, L., Yang, J., Kuang, K., Ni, B., Gao, Y., Sun, Y., Gao, P., Ma, W., Tan, M., Kang, H., Chen, J., Li, M.: Deep-learning-assisted detection and segmentation of rib fractures from ct scans: development and validation of fracnet. eBioMedicine **62**, 103106 (2020). https://doi.org/10.1016/j.ebiom.2020.103106
32. Yang, X., Xia, D., Kin, T., Igarashi, T.: Intra: 3D intracranial aneurysm dataset for deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
33. Ying, C., et al.: NAS-Bench-101: towards reproducible neural architecture search. arXiv (2019)
34. Gülcü, A., Kuş, Z.: Neural Architecture Search Using Differential Evolution in MAML Framework for Few-Shot Classification Problems, pp. 143–157. Springer, Berlin (2023)
35. Wei, J.: Genetic u-net: automatically designed deep networks for retinal vessel segmentation using a genetic algorithm. IEEE Trans. Med. Imaging **41**(2), 292–307 (2022)
36. Awad, N., Mallik, N., Hutter, F.: Differential evolution for neural architecture search. arXiv preprint arXiv:2012.06400 (2020)
37. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-based differential evolution. IEEE Trans. Evol. Comput. **12**(1), 64–79 (2008). https://doi.org/10.1109/TEVC.2007.894200
38. Kuş, Z., Kiraz, B., Göksu, T.K., Aydın, M., Özkan, E., Vural, A., Kiraz, A., Can, B.: Differential evolution-based neural architecture search for brain vessel segmentation. Eng. Sci. Technol. Int. J. **46**, 101502 (2023). https://doi.org/10.1016/j.jestch.2023.101502
39. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341 (1997)
40. Xie, X., Song, X., Lv, Z., Yen, G.G., Ding, W., Sun, Y.: Efficient evaluation methods for neural architecture search: a survey (2023). arXiv:https://arxiv.org/abs/2301.05919
41. Kuş, Z., Aydin, M., Kiraz, B., Kiraz, A.: Neural architecture search for biomedical image classification: a comparative study across data modalities. Artif. Intell. Med. **160**, 103064 (2025). https://doi.org/10.1016/j.artmed.2024.103064
42. Song, X., Xie, X., Lv, Z., Yen, G.G., Ding, W., Lv, J., Sun, Y.: Efficient evaluation methods for neural architecture search: a survey. IEEE Trans. Artif. Intell. **5**(12), 5990–6011 (2024). https://doi.org/10.1109/TAI.2024.3477457
43. Salmani Pour Avval, S., Eskue, N.D., Groves, R.M., Yaghoubi, V.: Systematic review on neural architecture search. Artif. Intell. Rev. (2025). https://doi.org/10.1007/s10462-024-11058-w
44. Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast bayesian optimization of machine learning hyperparameters on large datasets. In: Singh, A., Zhu, J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, 54, pp. 528–536. PMLR (2017)