MedPipe: End-to-End Joint Search of Data Augmentation and Neural Architecture for 3D Medical Image Classification

1st Xin HE
Dept. Computer Science
Hong Kong Baptist University
Hong Kong, China
csxinhe@comp.hkbu.edu.hk

2nd Xiaowen CHU
Data Science and Analytics Thrust
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
xwchu@ust.hk

***Abstract*—Data augmentation plays a crucial role in deep learning-based medical imaging analysis, but manually designing tailored data augmentation strategies for each dataset is impractical. Although automatic data augmentation (ADA) techniques have been explored, they often focus solely on data augmentation without considering the importance of neural architecture. Similarly, neural architecture search (NAS) methods mainly concentrate on optimizing the neural architecture while overlooking the impact of data augmentation. However, both data augmentation and neural architecture are interrelated and should be considered together. The joint optimization of data augmentation and neural architecture can lead to improved model performance by harnessing the complementary effects of customized data augmentation strategies and compatible neural architectures. Despite this, the seamless integration of data augmentation and neural architecture search remains underexplored. To address this research gap, we propose *MedPipe*, an approach that enables end-to-end joint search of data augmentation and neural architecture. We introduce a compact data augmentation search space and unify data augmentation and neural architecture into a cohesive network. This allows simultaneous exploration, optimizing their synergy for enhanced performance. Experimental evaluation on nine medical datasets highlights the necessity of the joint search for data augmentation and neural architecture, demonstrating the superior performance of our approach. Our work opens up possibilities for future applications in diverse medical domains.**

***Index Terms*—Neural Architecture Search (NAS), Automatic Data Augmentation (ADA), Medical Image Classification**

I. Introduction

Medical image classification plays a vital role in various healthcare applications. Deep learning (DL) methods have shown promising results in this field, but they often necessitate high-quality and large-scale datasets. Unfortunately, acquiring such datasets can be challenging due to the privacy concern, leading to the limitations and imbalance in data availability and labeling. Consequently, researchers have endeavored to design a set of data augmentation operations, to increase the size and diversity of the training data [1], [2], thus leading to better model performance. However, medical imaging data is known for its heterogeneity, which means that different datasets, even within the same domain, require tailored data augmentation. Notably, diverse diseases exhibit distinctive characteristics in terms of organ or lesion features, manifesting substantial variations. Furthermore, even within the same disease category, variations stemming from imaging devices and racial disparities contribute to dataset discrepancies. Designing data augmentation strategies manually for every dataset is not only time-consuming but also impractical.

Several studies have explored the use of automatic data augmentation (ADA) techniques to search for effective data augmentation (DA) strategies in medical tasks, yielding promising results [3], [4]. However, these studies often focus solely on optimizing data augmentation, overlooking the importance of considering compatible model architectures. Similarly, neural architecture search (NAS) has been widely employed to discover optimal model architectures for various tasks [5], [6]. However, most NAS studies overlook the significance of incorporating data augmentation during the search process.

The separation of data augmentation and neural architecture optimization can limit the exploration of their interdependence and potential synergy. By jointly optimizing data augmentation and neural architecture, researchers can leverage the complementary effects of tailored data augmentation strategies and compatible neural architectures. For example, when dealing with a small and imbalanced medical dataset, it is advisable to employ a relatively simple data augmentation strategy and neural architecture to mitigate overfitting. Conversely, a larger and imbalanced dataset may require a more complex model with an appropriate data augmentation strategy to address underfitting issues.

The research gap lies in the lack of studies that seamlessly integrate data augmentation and neural architecture search. While previous research [7], [8] has achieved improvements by optimizing each component separately, exploring their interplay can potentially lead to even better performance and more efficient solutions. Therefore, it is essential to investigate the joint optimization of data augmentation and neural architecture

Xiaowen Chu is with Data Science and Analytics Thrust, The Hong Kong University of Science and Technology (Guangzhou) (email: xwchu@ust.hk). Corresponding author.

to fully exploit their interrelated nature.

In this work, we propose *MedPipe*, a framework that enables the end-to-end joint search of data augmentation and neural architecture. By introducing a compact data augmentation search space and unifying data augmentation and neural architecture as a cohesive network, we facilitate simultaneous exploration and optimization of both components. Our evaluation on nine medical datasets demonstrates the necessity and effectiveness of jointly searching data augmentation and neural architecture, showcasing the superior performance of our approach. Our work bridges the gap between data augmentation and neural architecture search, paving the way for future applications in diverse medical domains.

We evaluate our method on nine publicly available 3D medical datasets for image classification tasks, demonstrating the superior performance of our approach compared to state-of-the-art methods. Our results show that *MedPipe* can effectively discover tailored data augmentation and neural architecture, leading to improved model performance. In summary, our work makes the following contributions:

- We propose *MedPipe*, a novel method for end-to-end joint search of data augmentation and neural architecture, which addresses the interdependence between the two aspects.
- We introduce a compact data augmentation search space and unify the concepts of data augmentation and neural architecture, enabling simultaneous exploration and optimization of both aspects.
- We evaluate *MedPipe* on nine public diverse medical datasets, demonstrating its superior performance compared to state-of-the-art methods.

II. Related Work

A. Manual CNNs for 3D Medical Image Classification

The existing DL-based methods for 3D medical image classification can be broadly divided into three classes according to the types of architectures: 2D, 2.5D, and 3D CNNs. The 2D CNN-based methods [9], [10] treat the 3D volumetric data as a sequence of 2D slices and use 2D CNN to extract features slice-by-slice and then fuse these features to make classifications. The 2.5D CNN is to feed 2D CNN with multiple angled slices from the 3D-space [11], [12] or with tri-slice data (a center slice with its two neighbor slices forming a normal three-channel RGB image) [13]. 2D and 2.5D CNNs are merely remedies for the lack of exploration of 3D context information. To fully use 3D medical data, researchers have attempted to use 3D CNNs. One of the most straightforward methods is to convert the existing 2D CNNs to 3D CNNs by replacing 2D operations (e.g., convolution and batch norm) with 3D ones. However, it is difficult to train the converted 3D CNNs due to the lack of large-scale 3D datasets for pretraining; thus, several methods propose to preserve the pretrained weights of 2D CNNs to 3D CNNs [14], [15].

B. Neural Architecture Search (NAS)

Neural architecture search (NAS) [5], [6] has been widely used to automatically search superior models and has achieved remarkable results on various tasks with natural images [16]–[19]. NAS has three mainstream search methods: reinforcement learning (RL) [16], [18], evolutionary algorithm (EA) [19], and gradient-based methods [17], [20]. Recently, many works have applied NAS to search CNNs for 3D medical images, most of them using EA [21]–[23], i.e., maintaining and mutating a set of architectures to produce better offsprings based on validation performance. However, the EA-based methods require many computational resources, especially for 3D medical images, e.g., [22] took 5 days with 64 GPUs. To reduce the search cost of EA, [23] proposes first to perform evolution on 2D models and then convert the searched 2D models to 3D models by replacing 2D convolutions with 3D convolutions, but this inevitably introduces inconsistencies between searched and final derived models. Gradient-based NAS methods [24]–[27] relax the discrete architectures into continuous representations and allow to search architectures in a differentiable way. In our work, we also use the gradient-based method and incorporate Gumbel-Softmax technique [20] to sample a single network at a time, which can greatly reduce the search cost and improve search efficiency.

C. Automatic Data Augmentation (ADA)

Due to the success of NAS in improving model performance, data augmentation is often overlooked. Most works usually apply a simple hand-crafted data augmentation that comprises, for example, only rotation and flipping. However, different datasets usually require different data augmentation. A clear example is that vertical flipping may mislead model training in the digit recognition task. In [28], the authors systematically compare different data augmentation operations applied to the 3D brain tumor scan and conclude that brightness and elastic augmentations can improve model training while flipping would not help much. Inspired by NAS, many automatic data augmentation (ADA) methods have been proposed to design specific data augmentation for different datasets without requiring extensive prior knowledge. The early ADA methods are applied to 2D natural images [29]–[35]. Some recent works [36]–[38] have extended ADA to 3D medical images. In [36], the authors use RL to search only the probability of augmentation operations. In [37], the authors further incorporate both the probability and the interval of the magnitude of augmentation operations into the search space.

III. Method

In Sec. III-A, we first present our joint search space, where each possible network is a combination of unified data augmentation (DA) and neural architecture (NA). Then, Sec. III-B describes the single-path based differentiable search algorithm on finding the optimal unified network.

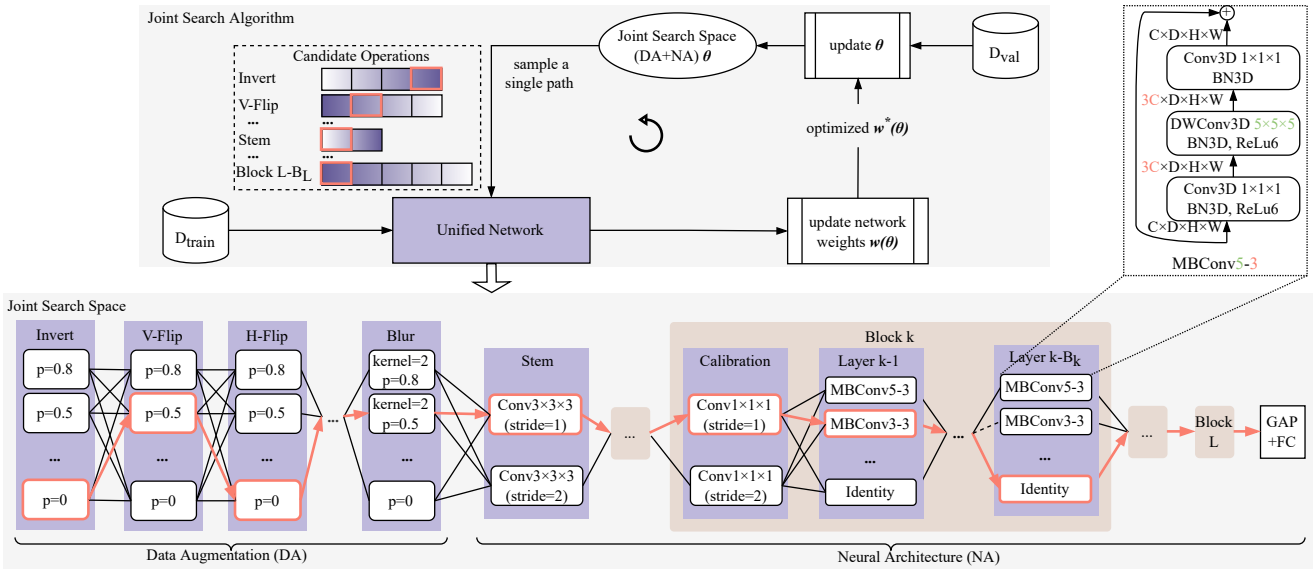


Fig. 1. The overall pipeline of *MedPipe*. In each iteration, the joint search algorithm first randomly samples a single path, i.e., one operation from each type of operation set, and then iteratively optimizes the weights of the sampled network $w(\theta)$ and distribution θ in a differentiable manner. The sampling probability of each operation is determined by the search space distribution θ, with darker color indicating a higher sampling probability. Data augmentation (DA) and neural architecture (NA) are combined into a unified network in our joint search space. All possible unified networks are subnets of a supernet and share weights with each other, and the path along the red arrows indicates the currently sampled subnet. An example of MBConv5-3 is given in the upper right corner, where *DWConv3D* denotes 3D depthwise convolution, *BN3D* denotes 3D batch normalization, and C, D, H, W indicate channel, slice number, height, and width. (Best viewed in color.)

A. Joint Search Space

As Fig. 1 shows, our unified network combines data augmentation and neural architecture. For ease of explanation, we introduce them separately.

1) Data Augmentation Space: The previous automatic data augmentation (ADA) methods [29], [31], [32]search for a set of sub-policies, each containing a fixed number of data augmentation operations, and different sub-policies may contain repeated operations. To avoid redundancy, we design a novel and compact data augmentation search space. Specifically, as shown in Fig. 1, the data augmentation part of the unified network comprises multiple layers, and each layer belongs to a different data augmentation type and has multiple predefined operations.

Candidate Data Augmentation operations: Table I details 12 types of operations used in our work, where *V/H/D-Rotation* and *V/H/D-Flip* are six different types. A data augmentation $\{o_i\}_{i=1}^{12}$ is formed by selecting one operation from each layer. Each operation o_i is associated with two variables, the probability of applying the operation p and the magnitude μ. We set 3 probabilities for each operation, $\{0, 0.5, 0.8\}$. Some operations have no magnitude, e.g., *Invert*, *Gaussion noise*, and *Flip*. The kernel size of *Blur* has three discrete magnitude choices $\{2, 3, 5\}$. For those operations with continuous magnitude, we predefine the range of left boundary (LB) and right boundary (RB), each of which is discretized into 5 values with uniform spacing. For example, there are $5 \times 5 \times 4$ possible *V-Rotation* operations, and suppose the searched LB and RB value is -15 and 30, then a random degree from the uniform distribution of [-15,30] will be drawn for vertical rotation. In summary, an operation o_i to an image x is denoted by $\hat{x} = \hat{o}_i(x) = b \cdot o_i(x; \mu_i, p_i) + (1 - b) \cdot x$, where $b \in \{0, 1\}$ is sampled from the Bernoulli distribution, i.e., $b = 1$ with the probability p_i.

2) Neural Architecture Space: The neural architecture part of the unified network comprises a stem layer, L different searchable blocks, and a 3D global average pooling (GAP) layer [39] followed by a fully-connected (FC) layer. The k-th searchable block comprises a calibration layer and B_k normal layers, where $k \in \{1, ..., L\}$. In [40], the stem layer and each calibration layer contain a fixed 3D convolution with predefined kernel size and stride value. The stride can control the spatial resolution changes. However, this predefined network pattern limits the diversity of architectures. Thus, in this work, we make the stem layer and all calibration layers searchable by inserting two candidate 3D convolution operations with stride sizes of 1 and 2, respectively.

Candidate Neural Architecture operations: All normal layers have 8 candidate operations, each with a stride of 1: {*MBConv3-3, MBConv5-3, MBConv7-3, MBConv3-4, MBConv5-4, MBConv7-4, MBConv3-6, Identity*}. *MBConvk-e* is the 3D version of the mobile inverted bottleneck convolution module [41], where k is the kernel size of intermediate convolution and e is the expansion ratio between the input channels and the inner channels. As shown in Fig. 1 (top-right), *MBConvk-e* comprises three sub-modules: 1) a 3D point-wise convolution with $1 \times 1 \times 1$ kernel size; 2) a 3D depthwise convolution with $k \times k \times k$ kernel size; 3) another 3D point-wise convolution with $1 \times 1 \times 1$ kernel

TABLE I
THE 3D DATA AUGMENTATION OPERATIONS. † V, H, AND D INDICATE VERTICAL, HORIZONTAL, AND DEPTH DIRECTIONS, RESPECTIVELY.

Type	DA Operations	Arguments	Magnitude	
			Left	Right
None	Noise	-	-	-
	V/H/D-Flip†	-	-	-
Continuous	V/H/D-Rotation†	degree	$[-30, 0]$	$[0, 30]$
	Erase	scale	(0, 0.1]	(0.1, 0.3]
		ratio	(0, 1]	[0.3, 3]
	Affine	scale	(0, 1]	(1, 3]
Discrete	Blur	kernel size	$\{2, 3, 4\}$	
	Invert	max value	$\{0.25, 0.5, 0.75, 1\}$	
	ColorJitter	brightness	$\{0.3, 0.5\}$	
		contrast	$\{0.5, 0.8\}$	

size. All convolutions in *MBConv* are followed by a 3D batch normalization and a ReLu6 activation function [42], except for the last that has no ReLu6 activation.

Settings: In our experiments, the architecture has $L = 6$ searchable blocks, where the first five blocks contain 1 calibration layer and 4 normal layers, and the last block has 1 calibration layer and 1 normal layer. The number of output channels of the stem layer and 6 searchable blocks is 32 and $[32, 48, 64, 96, 160, 320]$, respectively.

3) Search Complexity: Let Ω be the joint search space that contains N sets of candidate operations of data augmentation and neural architecture, i.e., $\Omega = \{\mathcal{O}_i\}_{i=1}^{N}$, where $\mathcal{O}_1 = \mathcal{O}_{\text{V-Flip}}, \mathcal{O}_2 = \mathcal{O}_{\text{H-Flip}}, \cdots, \mathcal{O}_N = \mathcal{O}_{\text{Layer L-B}_L}$. For example, $\mathcal{O}_{\text{V-Flip}}$ is a set of 4 candidate *V-Flip* operations with different applying probabilities. In summary, we have 6.2×10^{15} possible data augmentation and 1.2×10^{21} possible NAs, resulting in 7.4×10^{36} possible unified networks.

B. Joint Search Algorithm

1) Bi-level Optimization: We formulate the joint search of data augmentation and neural architecture as a bi-level optimization problem, as Eq. 1.

$$\begin{aligned} &\min_{\theta} \mathcal{L}_{\text{val}}\ (\omega_\theta^*, \theta) \\ &\text{s.t. } \omega_\theta^* = \operatorname{argmin}_{\omega_0} \mathcal{L}_{\text{train}}\ (\omega_\theta, \theta) \end{aligned} \tag{1}$$

where the outer step aims to find the optimal unified network (data augmentation and neural architecture) encoded by θ on the validation set $\mathcal{L}_{\text{val}}$, conditioned on that the network weights w_θ have already been optimized on the training set $\mathcal{L}_{\text{train}}$ in the inner step.

2) Weight-sharing: The joint search space is very large, so we cannot enumerate all unified networks and train their weights from scratch by brute force. To reduce the search cost, we adopt the weight-sharing strategy [17], [18] to allow all possible unified networks to share weights among a supernet. Specifically, each possible unified network can be considered as a subnet (i.e., a single path in Fig. 1) of the supernet and can directly inherit the corresponding weights from the supernet for performance estimation without training from scratch. Thus, the above bi-level optimization can be solved by iteratively optimizing inner and outer steps. Specifically, at each iteration, we first sample a unified network and use a batch of training data to train its weights (i.e., inner step), and then we use a batch of validation data to optimize the encoding θ of the networks (i.e., outer step). Next, we describe how to sample a single network in a differentiable way.

3) Differentiable Sampling: We reformulate the discrete search space as a continuous joint categorical distribution that encodes data augmentation and neural architecture. A unified network can be considered as a sampling of the joint categorical distribution θ. Specifically, the i-th data augmentation or neural architecture layer has K_i candidate operations, i.e., $K_i = |\mathcal{O}_i|$, and each operation is associated with a sampling probability $\theta_{i,j}$, where $i \in [1, N], j \in [1, K_i]$. Thus, the task of searching a unified network is transformed into learning a set of continuous variables $\theta = \{\{\theta_1^j\}_{j=1}^{K_1}, \cdots, \{\theta_N^j\}_{j=1}^{K_N}\}$. We relax the categorical choice of a particular operation to a softmax over all candidate operations, i.e., the output of the i-th searchable layer is calculated as

$$\begin{aligned} \bar{o}(x) &= \sum_{j=1}^{K_i} P_{i,j} \cdot o_{i,j}(x) \\ \text{s.t. } P_{i,j} &= \frac{\exp(\theta_{i,j})}{\sum_{m=1}^{K_i} \exp(\theta_{i,m})} \end{aligned} \tag{2}$$

where x is the input, $o_{i,j}$ is the j-th operation in $\mathcal{O}_i$ with a sampling probability variable of $\theta_{i,j}$, and $P_{i,j}$ indicates the softmax sampling probability of $o_{i,j}$ over all operations in $\mathcal{O}_i$.

4) Single-path Sampling: In Eq. 2, the output of each searchable layer is the weighted average of all candidate operations, which will cause a linear increase in the requirement of computational resources with the number of candidate operations. A straightforward way to reduce the cost is to sample a single path at a time, i.e., sampling the operation with the maximum softmax sampling probability $P_{i,j}$ at each layer; however, this cannot back-propagate gradients through θ in Eq. 2. To solve this problem, we use the Gumbel-Softmax trick [20], [43] to reformulate Eq. 2 as Eq. 3.

$$\begin{aligned} \bar{o}(x) &= \sum_{j=1}^{K_i} Z_{i,j} \cdot o_j(x) \\ \text{s.t. } Z_{i,j} &= \frac{\exp\left(\left(\log(P_{i,j}) + G_{i,j}\right)/\tau\right)}{\sum_{m=0}^{K_i} \exp\left(\left(\log(P_{i,m}) + G_{i,m}\right)/\tau\right)} \end{aligned} \tag{3}$$

where $P_{i,j}$ is the softmax sampling probability of Eq. 2, $G_{i,j} = -log(-log(u_{i,j}))$ is a Gumbel noise, $u_{i,j}$ is a random variable of uniform distribution $[0, 1]$, and τ is the softmax temperature. When $\tau \to 0$, the sampling results approximate to the one-hot distribution. However, since the divisor τ cannot be 0, we adopt the straight-through trick as Eq. 4 to post-process $Z_{i,j}$ to obtain the exact one-hot distributions.

$$Z_i = \operatorname{argmax}(Z_i) - \operatorname{StopGrad}(Z_i) + Z_i \tag{4}$$

where argmax is to find the index of maximum value in Z_i and returns a one-hot vector with the dimension of Z_i, and

TABLE II
STATISTICS OF SIX MEDMNIST3D DATASETS.

Datasets	Data Modality	#Classes	Train/Validation/Test
Organ3D	Abdominal CT	11	972 / 161 / 610
Nodule3D	Chest CT	2	1,158 / 165 / 526
Fracture3D	Chest CT	3	1,027 / 103 / 240
Adrenal3D	Abdominal CT Shape	2	1,188 / 98 / 298
Vessel3D	Brain MRA Shape	2	1,335 / 192 / 382
Synapse3D	Electron Microscope	2	1,230 / 177 / 352

StopGrad is to stop computing gradient of Z_i by treating it as a constant vector. Fig. 2 gives an example of applying Eq. 3 and Eq. 4, where the original softmax distribution P_i ([0.6,0.3,0.1]) is transformed into the one-hot distribution Z_i ([1,0,0]), i.e., $\bar{o}(x) = o_1(x)$ with a probability of 0.6.

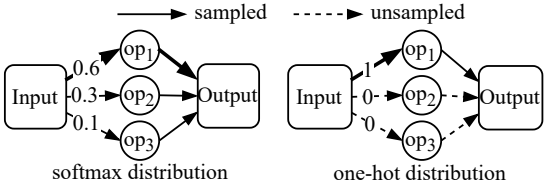


Fig. 2. An example of transforming the softmax distribution to the one-hot distribution.

IV. EXPERIMENTAL SETUP

This section first introduces the nine datasets used to verify the effectiveness of our framework, *MedPipe*. Then, the evaluation metrics and implementation setup are presented. The experimental results are given and analyzed in Sec. V.

A. Datasets

1) Six MedMNISTv2 3D Datasets: Table II gives statistics of the six 3D medical datasets provided by MedMNISTv2 [44], which cover diverse data modalities (e.g., CT, MRA, and electron microscope) and organs or tissues (e.g., nodule, fracture, adrenal, vessel, and synapse). The size of 3D volume data in each dataset is fixed to $28 \times 28 \times 28$.

2) Three COVID-19 3D CT Datasets: Table III shows the statistics of three COVID-19 3D CT datasets: Clean-CC-CCII [40], MosMed [45] and COVID-CTset [9], where NCP and CP indicate novel coronavirus pneumonia and common pneumonia, respectively. Clean-CC-CCII is a cleaned version of CC-CCII [46] by removing noisy data and correcting the order of slices. Next, we describe the preprocessing steps. In Clean-CC-CCII and COVID-CTset, each patient may have multiple 3D scan data. Thus, to avoid data leakage, we partition the data set based on the patient ID.

Slice Sampling: Each CT scan contains a different number of gray-scale slices. To ensure input data dimensional consistency, we propose two slice sampling strategies: *random sampling* and *symmetrical sampling*. Both supports up-sampling and down-sampling. Random sampling is applied to the training set, while symmetrical sampling is performed on the test set to avoid randomness in predictions. Symmetrical sampling is to sample from the middle to both sides at equal intervals based on slice IDs. The relative order between slices remains the same before and after sampling.

TABLE III
STATISTICS OF THREE COVID-19 CT SCAN DATASETS.

Dataset [Country]	Classes	#Patients		#Scans	
		Train	Test	Train	Test
Clean- [46] CC-CCII [China]	NCP	726	190	1213	302
	CP	778	186	1210	303
	Normal	660	158	772	193
	Total	2164	534	3195	798
MosMed [45] [Russia]	NCP	601	255	601	255
	Normal	178	76	178	76
	Total	779	331	779	331
COVID-CTset [9] [Iran]	NCP	80	15	202	42
	Normal	200	82	200	82
	Total	280	97	402	124

B. Evaluation Metrics

We use accuracy as our primary evaluation metric. Additionally, given the imbalanced nature of most of the datasets, we also use precision, sensitivity, and F1-score as evaluation metrics. F1-score is defined as the harmonic mean two negatively correlated metrics, precision and sensitivity. The final F1-score is calculated as a weighted sum of each class.

C. Implementation

Our framework contains two stages, search and evaluation, and is based on Pytorch [47]. The differentiable 3D data augmentation operations are implemented based on Kornia [48]. We conduct experiments with 1 NVIDIA V100 GPU.

1) Search stage: In the search stage, we split the training set equally into a training set and a validation set, which are used to optimize the inner and outer steps of Eq. 1, respectively. We use two Adam [49] optimizers to update θ (data augmentation and neural architecture) and w_θ (weights) of models, each with an initial learning rate of 1e-3 and 3e-4, respectively. We search 50 epochs for each experiment. After each epoch, we save the sampled model and its information (e.g., parameter size and validation accuracy).

2) Evaluation stage: We select top-5 networks according to their validation accuracy in the search stage and train these networks for several epochs in the evaluation stage. Finally, the best-performing network will be retrained for 200 epochs with the entire training set and then evaluated on the test set. We use the Adam [49] optimizer with an initial learning rate of 0.001. The cosine annealing scheduler [50] is applied to adjust the learning rate. We use Cross-entropy as the loss function in the search and evaluation stages.

D. Reproducibility

All implementations have been open-sourced in Github [1], including the configurations of search and evaluation stages and the weights of the best model for nine 3D medical datasets. All test set results are reproducible using the given pretrained

[1]https://github.com/marsggbo/hyperbox_app/tree/medmnist

weights. On the other hand, the search process carries a certain degree of randomness because we use the stochastic gradient descent strategy. Our configuration file also provides information such as seed to ensure that the search process is as reproducible as possible.

V. Experimental Results

We conduct two experiments to evaluate the capabilities of our MedPipe. In Sec. V-A, We first perform initial experiments on the six MedMNISTv2 datasets, which have small resolution and diverse data, as a quick validation of our approach. Then, we further validate the generalization of our method on three public COVID-19 CT datasets with larger resolution datasets.

A. Results on the six MedMNISTv2 3D Medical Datasets

In this subsection, we present extensive initial experiments on six small and diverse medical datasets. The resolution for each data sample is $28 \times 28 \times 28$. Most of these datasets are imbalanced. For instance, the proportion of the positive class are 76.85%, 87.83%, 73.01%, and 88.74% in Adrenal3D, Nodule3D, Synapse3D, and Vessel3D datasets, respectively. This means even if we predict all data samples as the positive class, we can still get a high accuracy, so we also consider F1-score as the evaluation metric.

The baseline models fall into two types: 1) Manual models including 2.5D, 3D, and ACS [14] variants of ResNet18 and ResNet50, where ACS can convert 2D models to 3D ones and maintain the pretrained weights; and 2) Automated models consisting of Auto-Sklearn [51] and AutoKeras [52]. We compare two settings for *MedPipe*:

- *MedPipe(Decoupled)* means data augmentation and neural architecture are searched separately, and it has two stages, where stage-1 fixes the architecture (i.e., ResNet18_2.5D) to search only data augmentation, and stage-2 fixed the searched data augmentation to search MBConv-based architecture;
- *MedPipe(Joint)* means data augmentation and neural architecture are searched jointly.

1) Baseline models: Table IV shows the model performance on each dataset and Table V presents the average performance on six MedMNISTv2 datasets. The results show that human-designed models, such as ResNet18+ACS and ResNet50+ACS, achieve competitive performance on most datasets. Intriguingly, they surpass their 2.5D and 3D counterparts, indicating that the choice of architecture is crucial for achieving high performance in medical image analysis tasks.

Automated machine learning techniques, such as Auto-Sklearn and AutoKeras, fail to discover superior models on these datasets, as their searched models fail to outperform human-designed models. This observation implies that finding a suitable model for these datasets is a challenging endeavor that demands careful consideration.

2) Decoupled Search: We compare two different settings for *MedPipe*. *MedPipe*(Decoupled) showed that optimizing data augmentation and neural architecture separately may lead to suboptimal performance. The first stage set ResNet18+2.5D as the base model and searches only for data augmentation. The results show an improved performance of ResNet18+2.5D on certain datasets, such as Nodule3D, Fracture3D, and Synapse3D. Especially for Nodule3D, the F1-score is improved by 8.4%. However, when continue searching for the MBConv-based neural architecture, the performance of the searched model with the searched data augmentation in stage-1 significantly decreases on all datasets. This may be due to the incompatibility between the searched data augmentation and neural architecture. In other words, the data augmentation that was optimized in the first stage of *MedPipe*(Decoupled) may not be well-suited for use with the MBConv-based neural architecture that was searched for in the second stage. This mismatch between the data augmentation and neural architecture can result in a suboptimal final model.

3) Joint Search: In contrast, *MedPipe*(Joint) jointly searches for data augmentation and neural architecture and achieves state-of-the-art performance on most datasets. The superior performance of *MedPipe*(Joint) can be attributed to the synergistic effect of searching for both data augmentation and neural architecture simultaneously. By doing so, the search algorithm can explore the combined search space more effectively, leading to the discovery of optimal configurations that might not be found when searching separately. This result highlights the importance of considering the interplay between data augmentation and neural architecture in the context of medical image analysis, as the optimal combination of these two components can significantly enhance model performance.

B. Results on the Three COVID-19 CT Datasets

The initial experiments on the six MedMNISTv2 datasets have demonstrated the effectiveness and necessity of joint search of data augmentation and neural architecture. However, since the resolution of these datasets is quite small, further verification of the proposed *MedPipe* approach is required. To this end, we conducted experiments on three public COVID-19 CT datasets with larger resolutions to evaluate the generalizability of our approach.

These three datasets vary in resolution, including the number of slices in each 3D scan data, and the height and width of the scan data. Therefore, we conducted two preliminary experiments to investigate 1) the effect of the number of slices; and 2) the performance of a series of 2D and 3D human-designed models, including DenseNet121 [53], DenseNet201 [53], ResNet50 , ResNet101 and ResNeXt101 [54]. The 3D CNNs include R2Plus1D [55], MC3_18 [55], DenseNet3D121 [56], ResNet3D18/101 [56], PreActRes3D101 [56], ResNeXt3D101 [56]. These two experiments are implemented on Clean-CC-CC and apply the same data augmentations, including resize, center-crop, and normalization. Each model is trained for 200 epochs via the Adam [49] optimizer. The weight decay is 5e-4, and the initial learning rate is 0.001.

1) Preliminary Experiments: **Effect of Slice Number.** We select MC3_18, ResNet3D101, and DenseNet3D121, to evaluate their performance at different number of slices (i.e., 16, 32, 64, 128, and 256). From Fig. 3, we can see that

TABLE IV
PERFORMANCE RESULTS OF DIFFERENT MODELS ON VARIOUS MEDICAL DATASETS. DA AND NA INDICATE WHETHER DATA AUGMENTATION AND NEURAL ARCHITECTURE SEARCH WERE USED, RESPECTIVELY. ACC AND F1 REPRESENT ACCURACY AND F1-SCORE, RESPECTIVELY.

Model	DA	NA	Organ3D		Nodule3D		Fracture3D		Adrenal3D		Vessel3D		Synapse3D	
			ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
ResNet18+2.5D	×	×	0.788	0.790	0.903	0.833	0.451	0.396	0.772	0.686	0.846	0.856	0.696	0.705
ResNet18+3D	×	×	0.907	0.912	0.908	0.841	0.508	0.543	0.721	0.688	0.877	0.885	0.745	0.809
ResNet18+ACS [14]	×	×	0.900	0.907	0.910	0.863	0.497	0.545	0.754	0.772	0.928	0.923	0.722	0.742
ResNet50+2.5D	×	×	0.769	0.772	0.911	0.839	0.397	0.322	0.763	0.807	0.877	0.809	0.735	0.707
ResNet50+3D	×	×	0.883	0.888	0.910	0.859	0.494	0.554	0.745	0.704	**0.950**	0.909	0.795	0.808
ResNet50+ACS [14]	×	×	0.889	0.897	0.906	0.852	0.517	0.567	0.758	0.791	0.858	0.910	0.709	0.732
Auto-Sklearn [51]	×	✓	0.814	0.822	0.926	0.872	0.453	0.403	0.802	0.791	0.915	0.908	0.730	0.629
AutoKeras [52]	×	✓	0.804	0.816	0.902	0.629	0.458	0.446	0.705	0.783	0.894	0.787	0.724	0.073
MedPipe(Decoupled) Stage-1	✓	×	0.759	0.748	**0.918**	0.917	0.525	0.452	0.745	0.731	0.829	0.843	0.761	0.757
MedPipe(Decoupled) Stage-2	×	✓	0.949	0.952	0.796	0.821	0.566	0.568	0.792	0.668	0.887	0.834	0.733	0.623
MedPipe(Joint Search)	✓	✓	**0.964**	**0.963**	0.913	**0.918**	**0.575**	**0.579**	**0.826**	**0.808**	**0.950**	**0.950**	**0.801**	**0.812**

TABLE V
AVERAGE PERFORMANCE ON SIX 3D MEDMNISTV2 DATASETS.

Models	DA	NA	AVG ACC	AVG F1
ResNet18+2.5D	×	×	0.743	0.711
ResNet18+3D	×	×	0.777	0.779
ResNet18+ACS [14]	×	×	0.785	0.792
ResNet50+2.5D	×	×	0.742	0.709
ResNet50+3D	×	×	0.796	0.787
ResNet50+ACS [14]	×	×	0.773	0.791
Auto-Sklearn [51]	×	✓	0.773	0.737
AutoKeras [52]	×	✓	0.747	0.589
MedPipe(Decoupled) Stage-1	✓	×	0.756	0.741
MedPipe(Decoupled) Stage-2	×	✓	0.787	0.744
MedPipe(Joint Search)	✓	✓	**0.838**	**0.838**

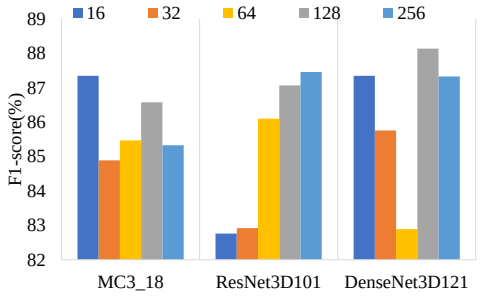


Fig. 3. The model performance (F1) at different number of slices.

the performance of ResNet3D101 improves as the number of slices increases, while the performance of MC3_18 and DenseNet3D121 fluctuates, which shows no significant relationship between the model performance and the number of slices.

Comparison between 2D and 3D CNNs. Since the previous experiment revealed no significant relationship between the number of slices and model performance, we fix the number of slices as 64 and set the height and width as 128×128. Table VI presents the performance comparison between 2D and 3D CNNs, indicating that 3D CNNs generally outperform 2D CNNs. Besides,it is observed that there is no apparent correlation between model size and performance, as ResNet3D18, despite being medium-sized, achieves the best results of accuracy, sensitivity, and F1-score.

TABLE VI
THE PERFORMANCE OF 3D AND 2D CNNS. ACC, PRE, AND SEN INDICATE ACCURACY, PRECISION, AND SENSITIVITY, RESPECTIVELY.

Model	Size (MB)	ACC	PRE	SEN	F1
R2Plus1D [55]	119.41	85.25	88.11	83.44	85.71
MC3_18 [55]	43.84	85.82	87.07	84.77	85.91
DenseNet3D121 [56]	43.06	83.83	**88.68**	77.81	82.89
ResNet3D18 [56]	126.53	**87.26**	86.18	**86.75**	**86.47**
ResNet3D101 [56]	325.21	84.50	84.62	83.77	84.19
PreActRes3D101 [56]	325.19	82.96	80.32	82.45	81.37
ResNeXt3D101 [56]	1628.52	83.33	85.87	80.46	83.08
DenseNet121	30.44	79.50	85.06	73.51	78.86
DenseNet201 [53]	76.35	81.75	78.98	82.12	80.52
ResNet50	97.49	76.80	73.62	74.83	74.22
ResNet101	169.94	79.43	78.35	75.50	76.90
ResNeXt101 [54]	338.71	76.19	74.58	72.85	73.70

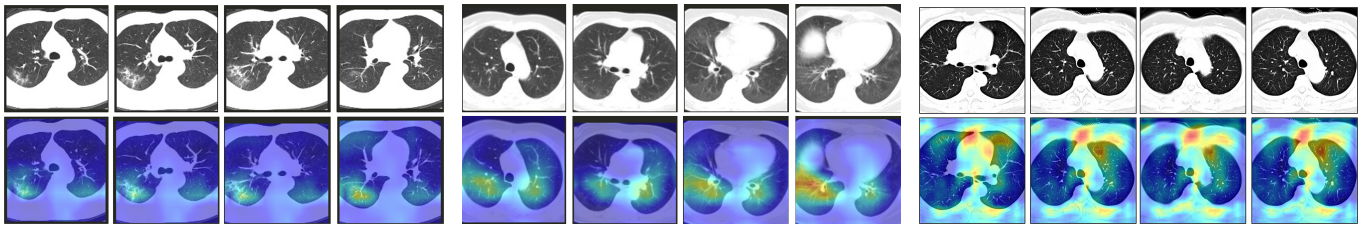
2) Performance Comparison: The above two preliminary experiments show that 1) *3D CNNs are potentially better than 2D CNNs to deal with 3D medical data*, and 2) *increasing the number of slices in 3D scan data does not necessarily improve the model performance*. Therefore, we focus on searching 3D networks on the three COVID-19 CT datasets and manualy set different scan sizes for each dataset to provide a comprehensive assessment of our *MedPipe*. Specifically, the scan sizes (slice number×height×width) of CC-CCII [46], MosMed [45], and COVID-CTset [9] are fixed to $32 \times 128 \times 128$, $40 \times 256 \times 256$, and $32 \times 512 \times 512$, respectively.

The performance comparison results are presented in Table VII. Our searched models outperform previous models on the three datasets in terms of accuracy, precision, and F1 score. Particularly noteworthy is the significant improvement achieved by our *MedPipe* model compared to the previous NAS-based approach, CovidNet3D-L [40], which shares the same architecture search space. Specifically, on the Clean-CC-CCII dataset, our searched model demonstrates an improvement of approximately 5% in accuracy and F1 score, indicating that the joint search of data augmentation and neural architecture can further unleash the model's potential. However, it is observed that the accuracy on the MosMed dataset is lower compared to the other two datasets. This disparity can be attributed to the dataset's imbalance, with a

TABLE VII

THE EXPERIMENTAL RESULTS ON THREE COVID-19 CT DATASETS. THE THIRD AND FOURTH COLUMNS INDICATE WHETHER THE CORRESPONDING METHOD SEARCHES DATA AUGMENTATION (DA) AND NEURAL ARCHITECTURE (NA). ACC, PRE, AND SEN INDICATE ACCURACY, PRECISION, AND SENSITIVITY, RESPECTIVELY.

Dataset [Country]	Model	DA	NA	GPU Days	Model size (MB)	ACC	PRE	SEN	F1
Clean-CC-CCII [China]	ResNet3D101 [56]	×	×	-	325.21	0.855	0.896	0.771	0.829
	DenseNet3D121 [56]	×	×	-	43.06	0.870	0.889	0.828	0.857
	MC3_18 [56]	×	×	-	43.84	0.862	0.871	0.828	0.849
	COVID-AL [57]	×	×	-	-	0.866	-	-	-
	Liu et al. [58]	×	×	-	-	0.881	0.840	0.892	0.865
	CovidNet3D-S [40]	×	✓	0.5	11.48	0.886	0.888	0.917	0.902
	CovidNet3D-L [40]	×	✓	1.3	53.26	0.887	0.905	0.881	0.893
	***MedPipe*(ours)**	✓	✓	1.5	27.38	**0.939**	**0.944**	**0.939**	**0.940**
MosMed [Russia]	ResNet3D101 [56]	×	×	-	325.21	0.818	0.813	0.972	0.886
	DenseNet3D121 [56]	×	×	-	43.06	0.795	0.842	0.922	0.880
	MC3_18 [56]	×	×	-	43.84	0.804	0.794	0.984	0.879
	DeCoVNet [59]	×	×	-	-	0.824	-	-	-
	CovidNet3D-S [40]	×	✓	0.2	12.48	0.812	0.788	**0.992**	0.878
	CovidNet3D-L [40]	×	✓	0.3	60.39	0.823	0.795	0.988	0.881
	***MedPipe*(ours)**	✓	✓	0.4	9.08	**0.849**	**0.887**	0.922	**0.904**
COVID-Ctset [Iran]	ResNet3D101 [56]	×	×		325.21	0.938	0.923	0.955	0.939
	DenseNet3D121 [56]	×	×	-	43.06	0.919	0.926	0.926	0.926
	MC3_18 [56]	×	×	-	43.84	0.926	0.909	0.945	0.927
	Vit-32×32 [60]	×	×	-	-	0.954	-	0.830	-
	AutoGluon Model [61]	×	✓	-	93.00	0.890	0.900	0.880	0.880
	CovidNet3D-S [40]	×	✓	0.2	8.36	0.943	0.927	0.905	0.916
	CovidNet3D-L [40]	×	✓	0.6	62.82	0.969	0.975	0.928	0.951
	***MedPipe*(ours)**	✓	✓	0.7	66.84	**0.976**	**0.976**	**0.964**	**0.969**



(a) Novel Coronavirus Pneumonia (NCP) (b) Common Pneumonia (CP) (c) Normal

Fig. 4. Examples of test scan data (upper row) and resulting Grad-CAM (lower row) from three classes in the Clean-CC-CCII dataset. Each 3D scan is presented by several 2D slices. Our model can effectively locate the lesion area of (a) NCP and (b) CP cases. For (c) the Normal case, the attention is on the non-lung area.

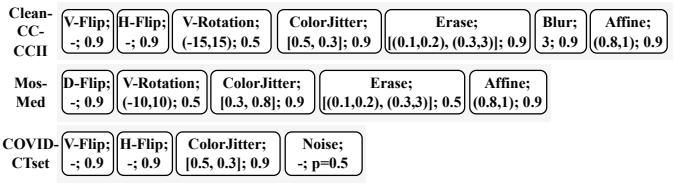


Fig. 5. The searched data augmentation policy on three COVID-19 datasets. Each operation is a triplet of type, magnitude, and probability.

ratio of approximately 3:1 between the number of samples in NCP and normal cases. Consequently, the models tend to overfit on the class with more data samples (i.e., NCP). Notably, while most previous models achieve extremely high sensitivity but relatively low precision, our joint search of data augmentation and neural architecture leads to the discovery of a model that strikes a balance between precision and sensitivity, resulting in the best F1 score of 90.38%.

Search Cost. To improve search efficiency, we halve the scan height and width in the search stage and restore them to the original size in the evaluation stage. As Table VII shows, compared with [40] that searches only neural architectures, our framework can complete the joint search of data augmentation and neural architecture with almost the same cost and find better models on all three COVID-19 datasets.

The Searched Unified Networks. Figure 5 illustrates the searched data augmentation for the three COVID-19 datasets. Operations with an applying probability of 0 are omitted. COVID-CTset exhibits a simpler data augmentation compared to the other two datasets due to its smaller size and balanced nature (refer to Table III). The data augmentation for COVID-CTset includes fewer operations, effectively mitigating the risk of overfitting. Notably, color jitter and flip operations with high applying probabilities are selected for all three data augmentation. Additionally, the searched stride values for the stem and the six calibration layers for Clean-CC-CCII,

MosMed, and COVID-CTset datasets are $[1, 2, 2, 2, 2, 2, 1]$, $[2, 2, 2, 2, 1, 2, 1]$, and $[2, 2, 2, 2, 1, 2, 1]$, respectively. Specifically, Clean-CC-CCII has a relatively small scan size, and the first stride of 1 helps prevent premature information loss. On the other hand, MosMed and COVID-CTset have larger scan sizes, and the first stride of 2 reduces computational costs and mitigates out-of-memory (OOM) errors that may occur due to the large scan data.

3) Feature Visualization: To verify the reliability of predictions made by our searched model, we utilize the MedCam library [62] to generate gradient-based class attention maps (Grad-CAM) [63]. Grad-CAM allows us to visualize the regions of the input data that have the greatest impact on the model's predictions. Figure 4 presents some examples of generated Grad-CAM maps from different classes in the Clean-CC-CCII dataset. In Figures 4 (a) and (b), our model effectively identifies lesion sites, such as areas with inflammatory infiltration and ground-glass opacity. In Figure 4 (c), the highlighted area is located outside the lung region since the normal class exhibits no abnormalities in the lungs.

VI. Conclusion

To the best of our knowledge, *MedPipe* represents a pioneering end-to-end framework that integrates the joint search of data augmentation (DA) and neural architecture (NA) for 3D medical image classification. A key contribution of our work is the modification of all 3D data augmentation operations to be differentiable, ensuring seamless integration within the search process. Moreover, in order to address computational constraints, we have devised a novel and compact joint search space that unifies DA and NA. The adoption of a single-path based differentiable search algorithm allows for the simultaneous search of DA and NA, further enhancing the efficiency of the framework. The extensive experimental evaluation conducted on nine 3D medical datasets, encompassing diverse data modalities, resolutions, imbalance ratios, and disease types, showcases the effectiveness and superiority of our *MedPipe* approach. *MedPipe* offers a comprehensive set of DA and NA components, providing researchers in the field of 3D medical imaging analysis, including segmentation and other related areas, with a valuable resource for further exploration and experimentation.

Acknowledgment

This work was partially supported by National Natural Science Foundation of China under Grant No. 62272122, a Hong Kong RIF grant under Grant No. R6021-20, and a Hong Kong CRF grant under Grant No. C2004-21GF.

References

[1] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential data augmentation techniques for medical imaging classification tasks," in *AMIA annual symposium proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 979.

[2] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big data*, vol. 6, no. 1, pp. 1–18, 2019.

[3] J. Xu, M. Li, and Z. Zhu, "Automatic data augmentation for 3d medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*. Springer, 2020, pp. 378–387.

[4] T. Qin, Z. Wang, K. He, Y. Shi, Y. Gao, and D. Shen, "Automatic data augmentation via deep reinforcement learning for effective kidney tumor segmentation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1419–1423.

[5] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *arXiv preprint arXiv:1808.05377*, 2018.

[6] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[7] T. Kashima, Y. Yamada, and S. Saito, "Joint search of data augmentation policies and network architectures," *arXiv preprint arXiv:2012.09407*, 2020.

[8] K. Zhou, L. Hong, S. Hu, F. Zhou, B. Ru, J. Feng, and Z. Li, "Dha: End-to-end joint optimization of data augmentation policy, hyper-parameter and architecture," *arXiv preprint arXiv:2109.05765*, 2021.

[9] M. Rahimzadeh, A. Attar, and S. M. Sakhaei, "A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/06/12/2020.06.08.20121541

[10] A. Shivdeo, R. Lokwani, V. Kulkarni, A. Kharat, and A. Pant, "Evaluation of 3d and 2d deep learning techniques for semantic segmentation in ct scans," in *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2021, pp. 1–8.

[11] H. R. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers, "A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2014, pp. 520–527.

[12] A. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. Van Riel, M. M. W. Wille, M. Naqibullah, C. I. Sánchez, and B. Van Ginneken, "Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.

[13] J. Ding, A. Li, Z. Hu, and L. Wang, "Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 559–567.

[14] J. Yang, X. Huang, Y. He, J. Xu, C. Yang, G. Xu, and B. Ni, "Reinventing 2d convolutions for 3d images," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 3009–3018, 2021.

[15] F. Gonda, D. Wei, T. Parag, and H. Pfister, "Parallel separable 3d convolution for video and volumetric data understanding," in *British Machine Vision Conference (BMVC)*, 2018.

[16] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=r1Ue8Hcxg

[17] H. Liu, K. Simonyan, and Y. Yang, "DARTS: differentiable architecture search," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=S1eYHoC5FX

[18] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 4092–4101. [Online]. Available: http://proceedings.mlr.press/v80/pham18a.html

[19] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 4780–4789. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33014780

[20] X. Dong and Y. Yang, "Searching for a robust neural architecture in four GPU hours," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 1761–1770.

[21] M. G. B. Calisto and S. K. Lai-Yuen, "Emonas: efficient multiobjective neural architecture search framework for 3d medical image segmentation," in *Medical Imaging 2021: Image Processing*, vol. 11596. International Society for Optics and Photonics, 2021, p. 1159607.

[22] Q. Yu, D. Yang, H. Roth, Y. Bai, Y. Zhang, A. L. Yuille, and D. Xu, "C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4126–4135.

[23] T. Hassanzadeh, D. Essam, and R. Sarker, "2d to 3d evolutionary deep convolutional neural networks for medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 712–721, 2020.

[24] Z. Zhu, C. Liu, D. Yang, A. Yuille, and D. Xu, "V-nas: Neural architecture search for volumetric medical image segmentation," in *2019 International conference on 3d vision (3DV)*. IEEE, 2019, pp. 240–248.

[25] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, "Dints: Differentiable neural network topology search for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5841–5850.

[26] S. Kim, I. Kim, S. Lim, W. Baek, C. Kim, H. Cho, B. Yoon, and T. Kim, "Scalable neural architecture search for 3d medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 220–228.

[27] X. Yang, Y. Huang, R. Huang, H. Dou, R. Li, J. Qian, X. Huang, W. Shi, C. Chen, Y. Zhang *et al.*, "Searching collaborative agents for multi-plane localization in 3d ultrasound," *Medical Image Analysis*, vol. 72, p. 102119, 2021.

[28] M. D. Cirillo, D. Abramian, and A. Eklund, "What is the best data augmentation for 3d brain tumor segmentation?" in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 36–40.

[29] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[31] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama, "Faster autoaugment: Learning augmentation strategies using backpropagation," in *European Conference on Computer Vision*. Springer, 2020, pp. 1–16.

[33] A. Liu, Z. Huang, Z. Huang, and N. Wang, "Direct differentiable augmentation search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 219–12 228.

[34] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2731–2741.

[35] R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama, "Meta approach to data augmentation optimization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2574–2583.

[36] D. Yang, H. Roth, Z. Xu, F. Milletari, L. Zhang, and D. Xu, "Searching learning strategy with reinforcement learning for 3d medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 3–11.

[37] J. Xu, M. Li, and Z. Zhu, "Automatic data augmentation for 3d medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 378–387.

[38] J. Lo, J. Cardinell, A. Costanzo, and D. Sussman, "Medical augmentation (med-aug) for optimal data augmentation in medical deep learning networks," *Sensors*, vol. 21, no. 21, p. 7018, 2021.

[39] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 2921–2929. [Online]. Available: https://doi.org/10.1109/CVPR.2016.319

[40] X. He, S. Wang, X. Chu, S. Shi, J. Tang, X. Liu, C. Yan, J. Zhang, and G. Ding, "Automated model design and benchmarking of deep learning models for covid-19 detection with chest ct scans," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 6, pp. 4821–4829, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16614

[41] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 4510–4520.

[42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. A. Mobilenets, "Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[43] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee

[44] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification," *arXiv preprint arXiv:2110.14795*, 2021.

[45] S. Morozov, A. Andreychenko, N. Pavlov, A. Vladzymyrskyy, N. Ledikhova, V. Gombolevskiy, I. Blokhin, P. Gelezhe, A. Gonchar, V. Chernina, and V. Babkin, "Mosmeddata: Chest ct scans with covid-19 related findings," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/05/22/2020.05.20.20100362

[46] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Zha, W. Liang, C. Wang, K. Wang *et al.*, "Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography," *Cell*, 2020.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[48] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, "Kornia: an open source differentiable computer vision library for pytorch," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3674–3683.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[50] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[51] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," *Advances in neural information processing systems*, vol. 28, 2015.

[52] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1946–1956.

[53] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 2261–2269. [Online]. Available: https://doi.org/10.1109/CVPR.2017.243

[54] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5987–5995. [Online]. Available: https://doi.org/10.1109/CVPR.2017.634

[55] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 6450–6459.

[56] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 6546–6555.

[57] X. Wu, C. Chen, M. Zhong, J. Wang, and J. Shi, "Covid-al: The diagnosis of covid-19 with deep active learning," *Medical Image Analysis*, vol. 68, p. 101913, 2021.

[58] X. Li, W. Tan, P. Liu, Q. Zhou, and J. Yang, "Classification of covid-19 chest ct images based on ensemble deep learning," *Journal of Healthcare Engineering*, vol. 2021, 2021.

[59] X. Wang, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and C. Zheng, "A weakly-supervised framework for covid-19 classification and lesion localization from chest ct," *IEEE transactions on medical imaging*, vol. 39, no. 8, pp. 2615–2625, 2020.

[60] J. C. Than, P. L. Thon, O. M. Rijal, R. M. Kassim, A. Yunus, N. M. Noor, and P. Then, "Preliminary study on patch sizes in vision transformers (vit) for covid-19 and diseased lungs classification," in *2021 IEEE National Biomedical Engineering Conference (NBEC)*. IEEE, 2021, pp. 146–150.

[61] T. Anwar, "Covid19 diagnosis using automl from 3d ct scans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 503–507.

[62] K. Gotkowski, C. Gonzalez, A. Bucher, and A. Mukhopadhyay, "M3d-cam: A pytorch library to generate 3d data attention maps for medical deep learning," 2020.

[63] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.