Automated Model Design and Benchmarking of Deep Learning Models for COVID-19 Detection with Chest CT Scans

**Xin He,[1] Shihao Wang,[1] Xiaowen Chu,[1]* Shaohuai Shi,[2] Jiangping Tang,[3] Xin Liu,[3]
Chenggang Yan,[3] Jiyong Zhang,[3]* Guiguang Ding[4]**

[1] Dept. Computer Science, Hong Kong Baptist University
[2] Dept. Computer Science and Engineering, Hong Kong University of Science and Technology
[3] School of Automation, Hangzhou Dianzi University [4] School of Software, Tsinghua University
chxw@comp.hkbu.edu.hk, jzhang@hdu.edu.cn

Abstract

The COVID-19 pandemic has spread globally for several months. Because its transmissibility and high pathogenicity seriously threaten people's lives, it is crucial to accurately and quickly detect COVID-19 infection. Many recent studies have shown that deep learning (DL) based solutions can help detect COVID-19 based on chest CT scans. However, most existing work focuses on 2D datasets, which may result in low quality models as the real CT scans are 3D images. Besides, the reported results span a broad spectrum on different datasets with a relatively unfair comparison. In this paper, we first use three state-of-the-art 3D models (ResNet3D101, DenseNet3D121, and MC3_18) to establish the baseline performance on three publicly available chest CT scan datasets. Then we propose a differentiable neural architecture search (DNAS) framework to automatically search the 3D DL models for 3D chest CT scans classification and use the Gumbel Softmax technique to improve the search efficiency. We further exploit the Class Activation Mapping (CAM) technique on our models to provide the interpretability of the results. The experimental results show that our searched models (CovidNet3D) outperform the baseline human-designed models on three datasets with tens of times smaller model size and higher accuracy. Furthermore, the results also verify that CAM can be well applied in CovidNet3D for COVID-19 datasets to provide interpretability for medical diagnosis. Code: https://github.com/HKBU-HPML/CovidNet3D.

Introduction

The Corona Virus Disease 2019 (COVID-19) pandemic is an ongoing pandemic caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The SARS-CoV-2 virus can be easily spread among people via small droplets produced by coughing, sneezing, and talking. COVID-19 is not only easily contagious but also a severe threat to human lives. The COVID-19 infected patients usually present pneumonia-like symptoms, such as fever, dry cough and dyspnea, and gastrointestinal symptoms, followed by a severe acute respiratory infection. The usual incubation period of COVID-19 ranges from one to 14 days. Many COVID-19 patients do not even know that they have been infected without any symptoms, which would easily cause delayed treatments and lead to a sudden exacerbation of the condition. Therefore, a fast and accurate method of diagnosing COVID-19 infection is crucial.

Currently, there are two commonly used methods for COVID-19 diagnosis. One is viral testing, which uses real-time reverse transcription-prognosis chain reaction (rRT-PCR) to detect viral RNA fragments. The other is making diagnoses based on characteristic imaging features on chest X-rays or computed tomography (CT) scan images. (Ai et al. 2020) conducted the effectiveness comparison between the two diagnosis methods and concluded that chest CT has a faster detection from the initial negative to positive than rRT-PCR. However, the manual process of analyzing and diagnosing based on CT images highly relies on professional knowledge and is time-consuming to analyze the features of the CT images. Therefore, many recent studies have tried to use deep learning (DL) methods to assist COVID-19 diagnosis with chest X-rays or CT scan images.

However, the reported accuracy of the existing DL-based COVID-19 detection solutions spans a broad spectrum because they were evaluated on different datasets, making it difficult to achieve a fair comparison. Besides, most studies focus on 2D CT datasets (Singh et al. 2020; Ardakani et al. 2020; Alom et al. 2020). However, the real CT scan is usually the 3D data. Thus it is necessary to use 3D models to classify 3D CT scan data. To this end, we use three state-of-the-art (SOTA) 3D DL models to establish the baseline performance on three open-source 3D chest CT scan datasets: CC-CCII[1] (Zhang et al. 2020b), MosMedData (Morozov et al. 2020) and COVID-CTset (Rahimzadeh, Attar, and Sakhaei 2020). The details are shown in Table 2.

In addition, designing a high-quality model for the specific medical image dataset is a time-consuming task and requires much expertise, which hinders the development of DL technology in the medical field. Recently, neural architecture search (NAS) has become a prevalent topic, as it can efficiently discover high-quality DL models automatically. Many studies have used the NAS technique to image classification and object detection tasks (Pham et al. 2018; Liu, Si-

*Corresponding author

[1] We find there are some errors and noises in the original dataset (Version 1.0). Therefore we built our version based on it.

Paper	Type	Open-source?	Dataset Statistics	Class Statistics (#slices)			Size of Test Set	Acc (%)
			#patients/#scans/#slices	NCP	Non-NCP			
					CP	Normal		
(Ghoshal et al.(2020)	X-ray(2D)	Yes	- / - / 5,941	68	4,290	1,583	1,188	88.39
(Zhang et al. 2020a)	X-ray(2D)	Yes	- / - / 1,531	100	1,431		764	-
Narin et al.(2020)	X-ray(2D)	Yes	- / - / 100	50	50		20	98.00
(Singh et al. 2020)	CT(2D)	No	- / - / 133	68	65		26	93.20
(Ardakani et al. 2020)	CT(2D)	No	194 / - / 1,020	510	510		102	99.63
(Alom et al. 2020)	CT(2D)	Yes	- / - / 425	178	247		45	98.78
(He et al. 2020)	CT(2D)	Yes	143 / - / 746	349	397		186	86.00
(Mobiny et al. 2020)	CT(2D)	Yes	- / - / 746	349	397		105	87.60
(Rahimzadeh et al.(2020)	CT(3D)	Yes	377 / 526 / 12,058	244‡	282‡		124‡	-
(Zheng et al. 2020)	CT(3D)	No	542 / 630 / -	313*	229*		131*	90.10
(Li et al. 2020)	CT(3D)	No	3,322 / 4,356 / -	1,296‡	1,735‡	1,325‡	427‡	-
(Morozov et al. 2020)	CT(3D)	Yes	1,110 / 1,110 / 46,411	856‡	254‡		331‡	-
(Zhang et al. 2020b)	CT(3D)	Yes	2,778 / 4,356 / 444,034	1,578‡	1,614‡	1,164‡	389‡	92.49

Table 1: Summary of the existing studies of DL-based methods for COVID-19 detection. NCP indicates the novel coronavirus pneumonia, Non-NCP includes CP (common pneumonia) and Normal. ‡: the number of scans. *: the number of patients.

monyan, and Yang 2019; Zoph et al. 2018; Tan et al. 2019). In this paper, we present a differentiable neural architecture search (DNAS) method combined with the Gumbel Softmax (Jang, Gu, and Poole 2017) technique to search neural architectures on three chest CT datasets: Clean-CC-CCII (Zhang et al. 2020b), MosMedData (Morozov et al. 2020), and COVID-CTset (Rahimzadeh, Attar, and Sakhaei 2020). We represent the search space by a supernet. Using the Gumbel Softmax technique, we can optimize only one subnetwork of the supernet at a time; therefore, the searching efficiency can be significantly improved, and the search stage can be finished in about 2 hours using 4 Nvidia Tesla V100 GPUs. We name the model searched by DNAS as **CovidNet3D**. The experimental results show that CovidNet3D can achieve comparable results to human-designed SOTA models with a smaller size. Furthermore, medical diagnoses generally require interpretability of the decision, so we apply Class Activation Mapping (CAM) (Zhou et al. 2016) techniques to provide interpretability for our CovidNet3D models. In summary, our contributions are summarized as follows:

- We use three manually designed 3D models to establish the baseline performance on three open-source COVID-19 chest CT scan datasets.
- To the best of our knowledge, we are the first to apply the differentiable NAS to search 3D DL models for COVID-19 chest CT scan datasets. Our DNAS framework can efficiently discover competitive neural architectures that outperform the baseline models on three CT datasets.
- We use the Class Activation Mapping (CAM) (Zhou et al. 2016) algorithm to add the interpretability of our DNAS-designed models, which can help doctors quickly locate the discriminative lesion areas on the CT scan images.

Related Work

In recent years, DL techniques have been proven to be effective in diagnosing diseases with X-ray and CT images (Litjens et al. 2017). To enable DL techniques to be applied in helping the detection of COVID-19, an increasing number of publicly available COVID-19 datasets have been proposed, as shown in Table 1.

Publicly-available Datasets of COVID-19

We separate the publicly available datasets into two different categories: the pre-pandemic datasets and the post-pandemic datasets which mainly differ in quality and quantity.

Pre-pandemic Datasets In the pre-pandemic period, the datasets for COVID-19 is very limited and low-quality. (Cohen, Morrison, and Dao 2020) provided a dataset by collecting chest X-ray and CT images of COVID-19 cases from public. But its quality has no guarantee since the images are not verified by medical experts. (Yang et al. 2020) is another CT dataset of COVID-19, which comprises CT images extracted from COVID-19 research papers. This dataset only contains 2D CT images because each patient has only one to several CT images instead of a complete 3D scan volume.

Post-pandemic Datasets With the rapid increase in the number of confirmed cases of COVID-19, many high-quality COVID-19 chest CT scan datasets have been provided, such as CC-CCII (Zhang et al. 2020b) and COVID-CTset (Rahimzadeh, Attar, and Sakhaei 2020). Some of them have annotations by doctors, e.g., COVID-19-CT-Seg-Dataset (Jun et al. 2020) and MosMedData (Morozov et al. 2020).

DL-based Methods for COVID-19 Detection

Many studies are conducted on CT images, but the 3D information of CT images is under-explored. In these studies (He et al. 2020; Mobiny et al. 2020; Singh et al. 2020), the authors only proposed 2D DL models for COVID-19 detection. (Ardakani et al. 2020) benchmarked ten 2D models on their private dataset with 102 testing images. On the other hand, the studies based on 3D CT images are relatively rare, mainly due to the lack of 3D COVID-19 CT scan datasets. (Li et al. 2020; Zheng et al. 2020) proposed 3D models with their private 3D CT datasets. There are also some studies conducted on X-ray images. For example, (Narin, Kaya, and Pamuk 2020) proposed three 2D DL models for COVID-19 detection. (Zhang et al. 2020a) introduced a deep anomaly detection model for fast and reliable screening. (Ghoshal and Tucker 2020) investigated the estimation of uncertainty and interpretability by Bayesian DL model on the X-ray images. (Alom et al. 2020) used both X-ray images and CT images to do segmentation and detection.

Neural Architecture Search

Recently, NAS has been applied into many tasks and achieved remarkable results (He, Zhao, and Chu 2021; Elsken, Metzen, and Hutter 2018). (Zoph and Le 2017; Zoph et al. 2018) used recurrent neural network (RNN) to generate neural architectures by using reinforcement learning (RL). Since then, several types of NAS methods have been proposed, such as evolutionary algorithm (EA) (Real et al. 2019), surrogate model-based optimization (SMBO) (Liu et al. 2018), and gradient descent (GD) based methods (Liu, Simonyan, and Yang 2019; Dong and Yang 2019). In (Dong and Yang 2019; Wu et al. 2019), the Gumbel Softmax (Jang, Gu, and Poole 2017) technique is incorporated to GD-based NAS, which significantly improve the searching efficiency.

Due to the success of NAS in natural image recognition (such as ImageNet (Deng et al. 2009)), researchers also apply it to the medical datasets. (Kim et al. 2019) applied NAS to segmentation task on Magnetic Resonance Imaging (MRI). (Faes et al. 2019) used Google Cloud AutoML platform to search and train models on five different medical datasets, and demonstrate that AutoML can generate competitive models comparable to human-designed models. However, there is no study applying the NAS technique to search 3D models for COVID-19 chest CT scan datasets. To this end, we exploit the NAS technique to three open-source COVID-19 CT datasets and discover high-quality 3D models outperforming human-designed 3D models.

Method

In this section, we first describe our search space for 3D CT scans classification models. Then, we introduce the differentiable neural architecture search (DNAS) method combined with the Gumbel Softmax technique (Jang et al. (2017).

Search Space

There are two critical points to be considered before designing the search space. One is that all datasets we use are composed of 3D CT scans; therefore, the searched model should be good at extracting the information from three-dimensional spatial data. The other is that the model should be lightweight, as the time required to process 3D data is much longer than 2D image data.

Although the cell-based search space (Pham et al. 2018; Liu, Simonyan, and Yang 2019) is one of the most commonly used search space, it has several problems: 1) the final model is built by stacking the same cells, which precludes the layer diversity; 2) many searched cells are very complicated and fragmented and are therefore inefficient for inference. MobileNetV2 (Sandler et al. 2018) is a lightweight model manually designed for mobile and embedded devices for efficient inference. Several NAS studies (Tan et al. 2019; Wu et al. 2019) have successfully used the layer modules (Sandler et al. 2018) including inverted residuals and linear bottlenecks to search neural architectures and achieved SOTA results on the 2D image datasets. Therefore, we use MobileNetV2 as a reference to design our 3D search space.

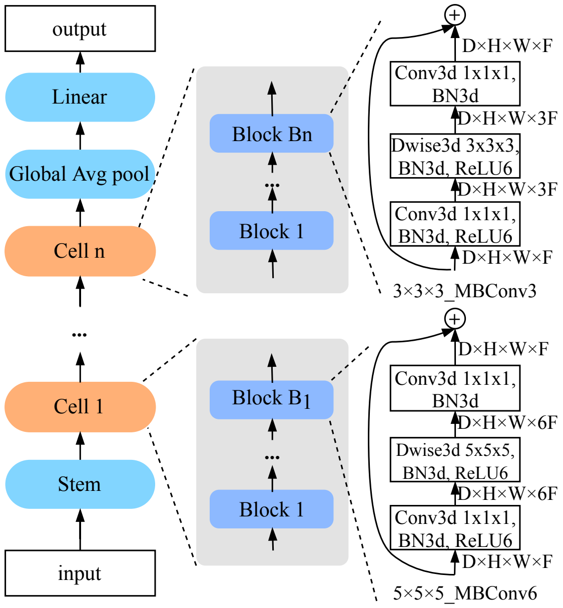

Figure 1: The overview of our search space. The model is generated by stacking a predefined number of cells. Each cell contains different number of blocks, and the block of different cells is different and needs to be searched. *Conv3d 1×1×1* denotes 3D convolution with $1\times1\times1$ kernel size, *Dwise3d* denotes 3D depthwise convolution, *BN3d* denotes 3D batch norm, $D \times H \times W \times F$ denotes tensor shape (depth, height, width, channel), and *MBConv* denotes mobile inverted bottleneck convolution.

As shown in Fig. 1, we represent the search space by a supernet, which consists of the stem layer, a fixed number of cells, and a linear layer. The stem layer performs convolutional operations, and the last linear layer follows behind a 3D global average pooling operation (Zhou et al. 2016). Each cell is composed of several blocks. The structures of all blocks need to be searched. In different cells, the number of channels and the number of blocks are different and hand-picked empirically. By default, all blocks have a stride of 1. However, if a cell's input/output resolutions are different, then its first block has a stride of 2. The blocks within the same cell have the same number of input/output channels. Inspired by MobileNetV2 (Sandler et al. 2018), each block is a MBConv-similar module (see Fig. 1). It consists of three sub-modules: 1) a point-wise $(1 \times 1 \times 1)$ convolution; 2) a 3D depthwise convolution with $K \times K \times K$ kernel size, where K is a searchable parameter; 3) another point-wise $(1 \times 1 \times 1)$ convolution. All convolutional operations are followed by a 3D batch normalization and a ReLU6 activation function (Howard et al. 2017), which is denoted by Conv3D-BN3D-ReLU6, and the last convolution has no ReLU6 activation. Another searchable parameter is the expansion ratio e, which controls the ratio between the size of the input bottleneck and the inner size. For example, $5 \times 5 \times 5$ *MBConv6* denotes that the kernel size of *MBConv* is $5 \times 5 \times 5$, and the expansion ratio is 6.

In our experiments, the search space is a fixed macro-architecture supernet consisting of 6 cells, where each has 4 blocks, but the last cell only has 1 block. We empirically collect the following set of candidate operations:

- $3 \times 3 \times 3$ *MBConv3*
- $3 \times 3 \times 3$ *MBConv4*
- $3 \times 3 \times 3$ *MBConv6*
- $5 \times 5 \times 5$ *MBConv3*
- $5 \times 5 \times 5$ *MBConv4*
- $7 \times 7 \times 7$ *MBConv3*
- $7 \times 7 \times 7$ *MBConv4*
- Skip connection

Therefore, it contains $8^{21} \approx 9.2 \times 10^{18}$ possible architectures. Finding an optimal architecture from such a huge search space is a stupendous task. We will introduce our search strategy in the following.

Differentiable NAS with Gumbel Softmax

According to (He, Zhao, and Chu 2021), gradient descent (GD) based NAS is an efficient method, and many studies use it to find competitive models with much shorter time and less computational resources (Dong and Yang 2019; Wu et al. 2019) than other NAS methods. Hence, in this paper, we use the GD-based method and combine it with the Gumbel Softmax (Jang, Gu, and Poole 2017) technique to discover models for COVID-19 detection.

Preliminary: DARTS DARTS (Liu, Simonyan, and Yang 2019) was one of the first studies to use GD-based method to search neural architectures. Each cell is defined as a directed acyclic graph (DAG) of N nodes, where each node is a network layer, and each edge between node i and node j indicates a candidate operation (i.e., block structure) that is selected from the predefined operation space $\mathcal{O}$. To make the search space continuous, DARTS (Liu, Simonyan, and Yang 2019) uses Softmax over all possible operations to relax the categorical choice of a particular operation, i.e.,

$$\begin{aligned} \bar{o}_{i,j}(x) &= \sum_{k=0}^{K} P_k o^k(x) \\ s.t.\ \ P_k &= \frac{\exp\left(\alpha_{i,j}^k\right)}{\sum_{l=0}^{K} \exp\left(\alpha_{i,j}^l\right)} \end{aligned} \quad , \tag{1}$$

where o^k indicates the k-th candidate operation performed on input x, $\alpha_{i,j}^k$ indicates the weight for the operation o^k between a pair of nodes (i, j), and K is the number of predefined candidate operations. The training and the validation loss are denoted by $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$, respectively. Therefore, the task of searching for architectures is transformed into a bilevel optimization problem of neural architecture α and the weights ω_α of the architecture:

$$\begin{aligned} \min_\alpha \quad & \mathcal{L}_{val}\left(\omega_\alpha^*, \alpha\right) \\ \text{s.t.} \quad & \omega_\alpha^* = \operatorname{argmin}_{\omega_\alpha}\ \mathcal{L}_{train}(\omega_\alpha, \alpha) \end{aligned} \tag{2}$$

Differentiable Model Sampling by Gumbel Softmax In DARTS, as Fig. 2 (left) shows, the output of each node is the weighted average of the mixed operation during the whole search stage. It causes a linear increase in the requirements of computational resources with the number of candidate operations. To alleviate this problem, we follow the same idea as (Dong and Yang 2019). Specifically, for each layer, only one operation is sampled and executed with the sampling probability distribution P_α defined in Equation 1. For example, the probability of being sampled for three operations in Fig. 2 (left) is 0.1, 0.2, and 0.7, respectively, but only one operation will be sampled at a time. Therefore, the sampling distribution P_α of all layers is encoded into a one-hot random distribution Z, e.g., $P_\alpha = [0.1, 0.2, 0.7] \rightarrow Z = [0, 0, 1]$.

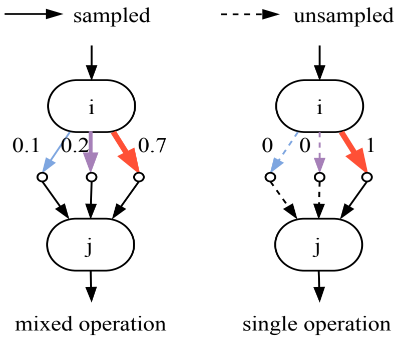

Figure 2: The comparison between two GD-based methods. (Left) Applying a mixture of all candidate operations, each with different weight. (Right) Only one operation is sampled at a time. Best viewed in color.

However, each operation is sampled from a discrete probability distribution Z, so we cannot back-propagate gradients through Z to α. To enable back-propagation, we use a reparameterization trick named Gumbel Softmax (Jang, Gu, and Poole 2017), which can be formulated by

$$Z_k = \frac{\exp\left(\left(\log \alpha_{i,j}^k + G_{i,j}^k\right)/\tau\right)}{\sum_{l=0}^{K} \exp\left(\left(\log \alpha_{i,j}^l + G_{i,j}^l\right)/\tau\right)} \quad , \tag{3}$$

where $G_{i,j}^k = -log(-log(u_{i,j}^k))$ is the k-th Gumbel sample, $u_{i,j}^k$ is a uniform random variable, and τ is the softmax temperature. When $\tau \rightarrow \infty$, the possibility distribution of all operations between each pair of nodes approximates to the one-hot distribution. To be noticed, we perform $argmax$ function on Equation 3 during the forward process but return the gradients according to the Equation 3 during the backward process.

Class Activation Mapping Algorithm

As mentioned above, the last linear layer follows behind a 3D global average pooling layer, which enables us to utilize class activation mapping (CAM) algorithm to generate 3D activation maps for our model. CAM exploits the global average pooling layer to calculate get the activation map M_c for class c, where each spacial element is given by

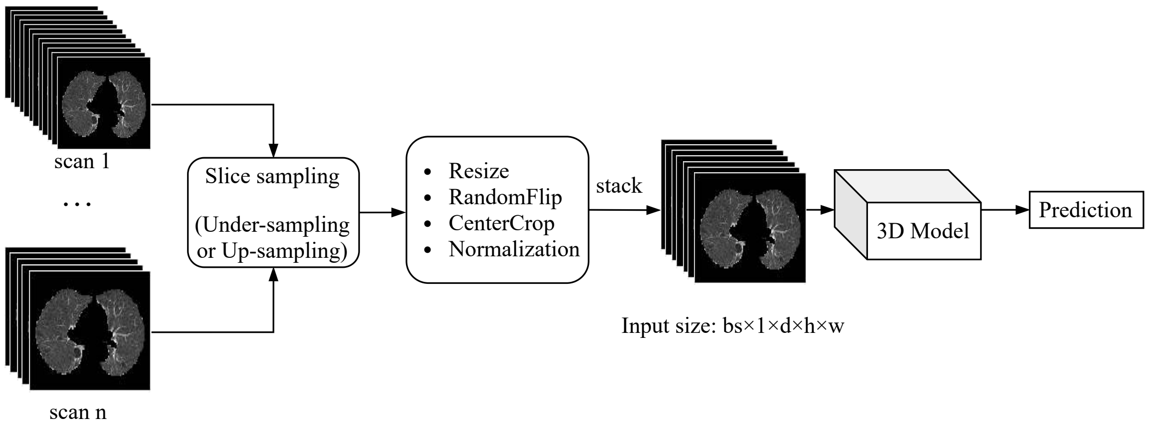

Figure 3: The pipeline of training 3D deep learning models. All CT scans need to be pre-processed by the slice sampling strategy to make sure that each scan contains the same number of slices. The input size of network is $bs \times 1 \times d \times h \times w$, where bs is batch size, d is the number of slices, h and w indicate the height and width, respectively.

$$M_c(x, y, z) = \sum_k w_k^c f_k(x, y, z) \tag{4}$$

where in a given image, $f_k(x, y, z)$ is the activation of unit k at the last convolutional layer before global average pooling layer at spatial location (x, y, z), w_k^c is the corresponding linear layer weight of class c for unit k. After getting the class activation map, we can simply upsample it to the size of the input scan images to visualize and identify the regions most relevant to the specific class.

Experiments

Datasets and Pre-processing

In this paper, we use three publicly available datasets: CC-CCII (Zhang et al. 2020b), MosMedData (Morozov et al. 2020) and COVID-CTset (Rahimzadeh, Attar, and Sakhaei 2020). Three datasets are all chest CT volumes. However, since the data format varies from three datasets, it is necessary to pre-process each dataset to make them follow a unified way of reading data.

The original CC-CCII dataset contains a total number 617,775 slices of 6,752 CT scans from 4,154 patients, but it has five main problems (i.e., damaged data, non-unified data type, repeated and noisy slices, disordered slices, and non-segmented slices) that would have high negative impacts on the model performance. To solve these problems, we manually remove the damaged, repeated and noisy data. Then we segment the lung part for the unsegmented slice image and convert the whole dataset to PNG format. After addressing the above problems, we build a clean CC-CCII dataset named **Clean-CC-CCII**, which consists of 340,190 slices of 3,993 scans from 2,698 patients.

Scan Images Construction Each CT scan contains a different number of slices, but DL models require the same dimensional inputs. To this end, we propose two slice sampling algorithms: *random sampling* and *symmetrical sampling*. Specifically, the random sampling strategy is applied to the training set, which can be regarded as the data augmentation, while the symmetrical sampling strategy is performed on the test set to avoid introducing randomness into the testing results. The symmetrical sampling strategy refers to sampling from the middle to both sides at equal intervals. The relative order between slices remains the same before and after sampling.

Dataset [Format]	Classes	#Patients		#Scans	
		Train	Test	Train	Test
Clean-CC-CCII [PNG]	NCP	726	190	1213	302
	CP	778	186	1210	303
	Normal	660	158	772	193
	Total	2164	534	3195	798
MosMedData [PNG]	NCP	601	255	601	255
	Normal	178	76	178	76
	Total	779	331	779	331
COVID-CTset [16bit TIFF]	NCP	80	15	202	42
	Normal	200	82	200	82
	Total	280	97	402	124

Table 2: The statistics of three CT scan datasets.

Benchmarking

We use three manually-designed 3D neural architectures as the baseline methods: DenseNet3D121 (Diba et al. 2017), ResNet3D101 (Tran et al. 2018), and MC3_18 (Tran et al. 2018). As shown in Fig. 3, after building the scan images by the sampling algorithm, we further apply transformations to scans, including resize, center-crop, and normalization. Besides, for the training set, we also perform a random flip operation in the horizontal or vertical direction. The other implementation details are as follows: we use the Adam (Kingma and Ba 2015) optimizer and the weight decay of 5e-4. We start the learning rate of 0.001 and anneal it down to 1e-5. All baseline models are trained for 200 epochs.

DNAS for CT Scan Classification

We apply the DNAS method combined with the Gumbel Softmax technique to search neural architectures on three datasets. The pipeline contains two sequential stages: search and evaluation, shown in Fig. 4.

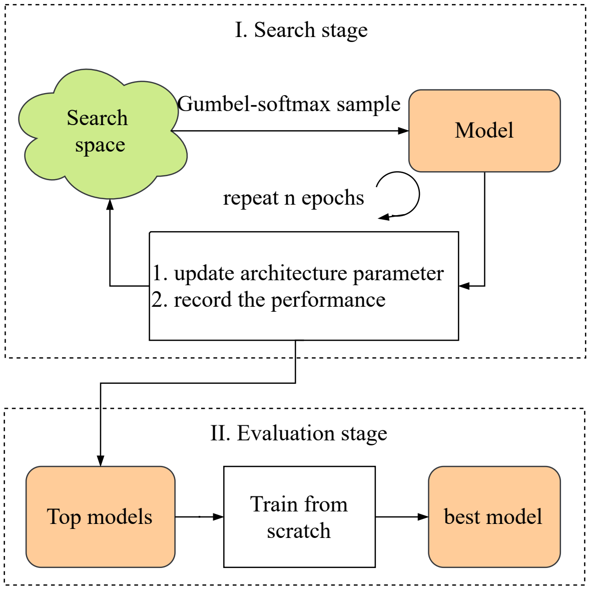

Figure 4: The pipeline of DNAS consists of two stages: the search and the evaluation stage.

Search Stage In our experiments, the supernet consists of 6 cells with the number of blocks of $[4, 4, 4, 4, 4, 1]$. Besides, the blocks within the same cell have the same number of channels. Here, we test two settings: small-scale and large-scale, where the number of channels of blocks in the 6 cells is $[24, 40, 80, 96, 192, 320]$ and $[32, 64, 128, 256, 512, 1024]$, respectively. We name the models searched under the two settings as **CovidNet3D-S** and **CovidNet3D-L**, respectively. The stem block is a Conv3D-BN3D-ReLU6 sequential module with the number of output channels fixed to 32.

To improve searching efficiency, we set the input resolution to 64×64, and the number of slices in a scan to 16. We implement three independent search experiments on three datasets. During the search stage, we split the training set into the training set $\mathcal{D}_\mathcal{T}$ and the validation set $\mathcal{D}_\mathcal{V}$. In each step, we first use $\mathcal{D}_\mathcal{V}$ to update the architecture parameters α, and then use the training set to update the sampled architecture weights ω_α. Besides, the architecture parameter α is optimized by the Adam (Kingma and Ba 2015) optimizer, and the architecture weights are optimized with the SGD optimizer with a momentum of 3e-4. The initial learning rate for both optimizers is 0.001. Each experiment is conducted on four Nvidia Tesla V100 GPUs (the 32GB PCIe version) and it can be finished in about 2 hours. After each epoch, we save the sampled architecture and its performance (e.g., accuracy). Therefore, we generate 100 neural architectures for each experiment after the search stage.

Evaluation Stage As Fig. 4 shows, the search stages records the performance of the sampled architectures. In the evaluation stage, we select top-10 architectures and training these architectures with the training set for several batches, then the best-performing architecture will be retrained for 200 epochs with the full training set, and then evaluated on the test set. We set different input resolutions for three datasets to evaluate the generalization of searched architectures. Besides, since the number of slices contained in CT scans of different datasets is different, we set the intermediate value for each dataset, shown in Table 3. Each evaluation experiment uses the same settings as follows: we use the Adam (Kingma and Ba 2015) optimizer with an initial learning rate of 0.001. The cosine annealing scheduler (Loshchilov and Hutter 2017) is applied to adjust the learning rate. We use Cross-entropy as the loss function.

Results and Analysis

Evaluation Metrics

We use four metrics to evaluate the model performance:

- $Precision = \frac{TP}{TP+FP}$
- $Sensitivity\ (Recall) = \frac{TP}{TP+FN}$
- $F1-score = \frac{2\times(Precision\times Recall)}{Precision+Recall}$
- $Accuracy = \frac{TN+TP}{TN+TP+FN+FP}$

The positive and negative cases are the COVID-19 class and the non-COVID-19 class, respectively. Specifically, TP and TN indicate the number of correctly classified COVID-19 and non-COVID-19 scans, respectively. FP and FN indicate the number of wrongly classified COVID-19 and non-COVID-19 scans, respectively. For the Clean-CC-CCII dataset, the non-COVID-19 class includes both normal and common pneumonia. The accuracy is the micro-averaging value for all test data to evaluate the overall performance of the model. Besides, we also take the model size as an evaluation metric to compare the model efficiency.

Results on Three CT Datasets

Table 3 divides the results according to the datasets. We can see that our searched CovidNet3D models outperform all baseline models on three datasets in terms of accuracy. Specifically, CovidNet3D-L models achieve the highest accuracy of three datasets. Besides, all CovidNet3D-S models are with much smaller sizes than the baseline models, but they can also achieve similar or even better results. For example, CovidNet3D-S (8.36 MB) achieves 94.27% accuracy on Covid-CTset, which is 41× smaller than ResNet3D101 (325.21 MB) with 0.4% higher accuracy. In summary, the results demonstrate that our DNAS method can discover well-performing models without inconsistency on network size, input size or scan depth (the number of slices).

We can also see that the performance of both baseline models and our CovidNet3D on the MosMedData dataset is not as good as that on the other two datasets. There are

Dataset	Model	Model size (MB)	Input size	#Slices	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score
Clean-CC-CCII	ResNet3D101	325.21	128×128	32	85.54	89.62	77.15	0.8292
	DenseNet3D121	43.06	128×128	32	87.02	88.97	82.78	0.8576
	MC3_18	43.84	128×128	32	86.16	87.11	82.78	0.8489
	CovidNet3D-S	11.48	128×128	32	88.55	88.78	**91.72**	**0.9023**
	CovidNet3D-L	53.26	128×128	32	**88.69**	**90.48**	88.08	0.8926
MosMedData	ResNet3D101	325.21	256×256	40	81.82	81.31	97.25	**0.8857**
	DenseNet3D121	43.06	256×256	40	79.55	**84.23**	92.16	0.8801
	MC3_18	43.84	256×256	40	80.4	79.43	98.43	0.8792
	CovidNet3D-S	12.48	256×256	40	81.17	78.82	**99.22**	0.8785
	CovidNet3D-L	60.39	256×256	40	**82.29**	79.50	98.82	0.8811
Covid-CTset	ResNet3D101	325.21	512×512	32	93.87	92.34	**95.54**	0.9392
	DenseNet3D121	43.06	512×512	32	91.91	92.57	92.57	0.9257
	MC3_18	43.84	512×512	32	92.57	90.95	94.55	0.9272
	CovidNet3D-S	8.36	512×512	32	94.27	92.68	90.48	0.9157
	CovidNet3D-L	62.82	512×512	32	**96.88**	**97.50**	92.86	**0.9512**

Table 3: The experimental results of standard human-designed models and DNAS-designed models.

two possible reasons. One is that the MosMedData datasets's original data format is NIfTI, but all our models do not converge when trained with NIfTI files; therefore we convert NIfTI to Portable Network Graphics (PNG) format, and this process would loss information of the input files. The other possible reason is that the MosMedData dataset is imbalanced (shown in Table 2), which increases the difficulty of model training.

We also find that the random seed greatly influences on the training of the searched CovidNet3D model through experiments. In other words, the results obtained by using different seeds for the same model would differ significantly. Hence, how to improve the robustness of NAS-based models is worthy for further exploring.

Interpretability

Although our model achieves promising result in detecting COVID-19 in CT images, classification result itself does not help clinical diagnosis without proving the inner mechanism which leads to the final decision makes sense. To inspect our CovidNet3D model's inner mechanism, we apply Class Activation Mapping (CAM) (Zhou et al. 2016) on it.

CAM is an algorithm that can visualize the discriminative lesion regions that the model focuses on. In Fig. 5, we apply CAM on each slice of a whole 3D CT scan volume from Clean-CC-CCII dataset. Regions appear red and brighter have a larger impact on the model's decision to classify it to COVID-19. From the perspective of the scan volume, we can see that some slices have more impacts on the model's decision than the others. In terms of a single slice, the areas that CovidNet3D focuses on has ground-glass opacity, which is proved a distinctive feature of CT images of COVID-19 Chest CT images (Bai et al. 2020). CAM enables the interpretability of our searched models (CovidNet3D), helping doctors quickly locate the discriminative lesion areas.

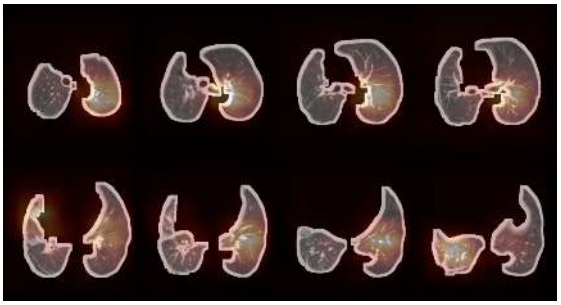

Figure 5: The class activation mappings generated on CovidNet3D on a chest CT scan of the Clean-CC-CCII dataset. Regions colored in red and brighter has more impact on model's decision to the class of COVID-19 while blue and darker region has less.

Conclusion

In this work, we introduce the differentiable neural architecture (DNAS) framework combined with the Gumbel Softmax technique to search 3D models on three open-source COVID-19 CT scan datasets. The results show that CovidNet3D, a family of models discovered by DNAS can achieve comparable results to the baseline 3D models with smaller size, which demonstrates that NAS is a powerful tool for assisting in COVID-19 detection. In the future, we will apply NAS to the task of 3D medical image segmentation to locate the lesion areas in a more fine-grained manner.

Acknowledgments

The research was supported by the grant RMGS2019_1_23 from Hong Kong Research Matching Grant Scheme . We would like to thank the anonymous reviewers for their valuable comments.

References

Ai, T.; Yang, Z.; Hou, H.; Zhan, C.; Chen, C.; Lv, W.; Tao, Q.; Sun, Z.; and Xia, L. 2020. Correlation of chest CT and RT-PCR Testing in Coronavirus Disease 2019 (COVID-19) in China: a Report of 1014 Cases. *Radiology* 200642.

Alom, M. Z.; Rahman, M. M. S.; Nasrin, M. S.; Taha, T. M.; and Asari, V. K. 2020. COVID_MTNet: COVID-19 Detection with Multi-Task Deep Learning Approaches. *arXiv preprint arXiv:2004.03747* .

Ardakani, A. A.; Kanafi, A. R.; Acharya, U. R.; Khadem, N.; and Mohammadi, A. 2020. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine* 121(March): 103795. ISSN 0010-4825. doi:https://doi.org/10.1016/j.compbiomed.2020.103795.

Bai, H. X.; Hsieh, B.; Xiong, Z.; Halsey, K.; Choi, J. W.; Tran, T. M. L.; Pan, I.; Shi, L.-B.; Wang, D.-C.; Mei, J.; et al. 2020. Performance of radiologists in differentiating COVID-19 from non-COVID-19 viral pneumonia at chest CT. *Radiology* 296(2): E46–E54.

Cohen, J. P.; Morrison, P.; and Dao, L. 2020. COVID-19 image data collection. *arXiv preprint 2003.11597* URL https://github.com/ieee8023/covid-chestxray-dataset.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 248–255. IEEE Computer Society. doi:10.1109/CVPR.2009.5206848.

Diba, A.; Fayyaz, M.; Sharma, V.; Karami, A. H.; Arzani, M. M.; Yousefzadeh, R.; and Van Gool, L. 2017. Temporal 3d convnets: New architecture and transfer learning for video classification. *arXiv preprint arXiv:1711.08200* .

Dong, X.; and Yang, Y. 2019. 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*, 1761–1770. Computer Vision Foundation / IEEE. doi:10.1109/CVPR.2019.00186.

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

Faes, L.; Wagner, S.; Fu, D.; Liu, X.; Korot, E.; Ledsam, J.; Back, T.; Chopra, R.; Pontikos, N.; Kern, C.; Moraes, G.; Schmid, M.; Sim, D.; Balaskas, K.; Bachmann, L.; Denniston, A.; and Keane, P. 2019. Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study. *The Lancet Digital Health* 1: e232–e242. doi:10.1016/S2589-7500(19)30108-6.

Ghoshal, B.; and Tucker, A. 2020. Estimating Uncertainty and Interpretability in Deep Learning for Coronavirus (COVID-19) Detection. *arXiv preprint arXiv:2003.10769* 1–14.

He, X.; Yang, X.; Zhang, S.; Zhao, J.; Zhang, Y.; Xing, E.; and Xie, P. 2020. Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans. *medRxiv* doi:10.1101/2020.04.13.20063941.

He, X.; Zhao, K.; and Chu, X. 2021. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems* 212: 106622. ISSN 0950-7051. doi:https://doi.org/10.1016/j.knosys.2020.106622.

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

Jang, E.; Gu, S.; and Poole, B. 2017. 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.

Jun, M.; Cheng, G.; Yixin, W.; Xingle, A.; Jiantao, G.; Ziqi, Y.; Minqing, Z.; Xin, L.; Xueyuan, D.; Shucheng, C.; Hao, W.; Sen, M.; Xiaoyu, Y.; Ziwei, N.; Chen, L.; Lu, T.; Yuntao, Z.; Qiongjie, Z.; Guoqiang, D.; and Jian, H. 2020. COVID-19 CT Lung and Infection Segmentation Dataset. *Zenodo* doi:10.5281/zenodo.3757476.

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

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Li, L.; Qin, L.; Xu, Z.; Yin, Y.; Wang, X.; Kong, B.; Bai, J.; Lu, Y.; Fang, Z.; Song, Q.; et al. 2020. Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. *Radiology* 200905.

Litjens, G.; Kooi, T.; Bejnordi, B. E.; Setio, A. A. A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J. A.; van Ginneken, B.; and Sánchez, C. I. 2017. A survey on deep learning in medical image analysis. *Medical Image Analysis* 42(1995): 60–88. doi:10.1016/j.media.2017.07.005.

Liu, C.; Zoph, B.; Neumann, M.; Shlens, J.; Hua, W.; Li, L.-J.; Fei-Fei, L.; Yuille, A.; Huang, J.; and Murphy, K. 2018. Progressive neural architecture search. *Proceedings of the European Conference on Computer Vision (ECCV)* 19–34.

Liu, H.; Simonyan, K.; and Yang, Y. 2019. DARTS: Differentiable Architecture Search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Loshchilov, I.; and Hutter, F. 2017. 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.

Mobiny, A.; Cicalese, P. A.; Zare, S.; Yuan, P.; Abavisani, M.; Wu, C. C.; Ahuja, J.; de Groot, P. M.; and Van Nguyen, H. 2020. Radiologist-Level COVID-19 Detection Using CT Scans with Detail-Oriented Capsule Networks. *arXiv preprint arXiv:2004.07407* .

Morozov, S.; Andreychenko, A.; Pavlov, N.; Vladzymyrskyy, A.; Ledikhova, N.; Gombolevskiy, V.; Blokhin, I.; Gelezhe, P.; Gonchar, A.; Chernina, V.; and Babkin, V. 2020. MosMedData: Chest CT Scans with COVID-19 Related Findings. *medRxiv* doi:10.1101/2020.05.20.20100362.

Narin, A.; Kaya, C.; and Pamuk, Z. 2020. Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks. *arXiv preprint arXiv:2003.10849* .

Pham, H.; Guan, M. Y.; Zoph, B.; Le, Q. V.; and Dean, J. 2018. Efficient Neural Architecture Search via Parameter Sharing. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 4092–4101. PMLR.

Rahimzadeh, M.; Attar, A.; and Sakhaei, S. M. 2020. A Fully Automated Deep Learning-based Network For Detecting COVID-19 from a New And Large Lung CT Scan Dataset. *medRxiv* doi:10.1101/2020.06.08.20121541.

Real, E.; Aggarwal, A.; Huang, Y.; and Le, Q. V. 2019. 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*, 4780–4789. AAAI Press. doi:10.1609/aaai.v33i01.33014780.

Sandler, M.; Howard, A. G.; Zhu, M.; Zhmoginov, A.; and Chen, L. 2018. 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*, 4510–4520. IEEE Computer Society. doi:10.1109/CVPR.2018.00474.

Singh, D.; Kumar, V.; Vaishali; and Kaur, M. 2020. Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. *European journal of clinical microbiology & infectious diseases : official publication of European Society of Clinical Microbiology* ISSN 0934-9723. doi:10.1007/s10096-020-03901-z.

Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; and Le, Q. V. 2019. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2820–2828. Computer Vision Foundation / IEEE. doi:10.1109/CVPR.2019.00293.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. 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*, 6450–6459. IEEE Computer Society. doi:10.1109/CVPR.2018.00675.

Wu, B.; Dai, X.; Zhang, P.; Wang, Y.; Sun, F.; Wu, Y.; Tian, Y.; Vajda, P.; Jia, Y.; and Keutzer, K. 2019. FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 10734–10742. Computer Vision Foundation / IEEE. doi:10.1109/CVPR.2019.01099.

Yang, X.; He, X.; Zhao, J.; Zhang, Y.; Zhang, S.; and Xie, P. 2020. COVID-CT-Dataset: A CT Scan Dataset about COVID-19. *arXiv preprint arXiv:2003.13865* .

Zhang, J.; Xie, Y.; Li, Y.; Shen, C.; and Xia, Y. 2020a. COVID-19 Screening on Chest X-ray Images Using Deep Learning based Anomaly Detection. *arXiv preprint arXiv:2003.12338* .

Zhang, K.; Liu, X.; Shen, J.; Li, Z.; Sang, Y.; Wu, X.; Zha, Y.; Liang, W.; Wang, C.; Wang, K.; et al. 2020b. Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using Computed Tomography. *Cell* .

Zheng, C.; bo Deng, X.; Fu, Q.; Zhou, Q.; Feng, J.; Ma, H.; Liu, W.; and Wang, X. 2020. Deep Learning-based Detection for COVID-19 from Chest CT using Weak Label. *medRxiv* .

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2016. 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*, 2921–2929. IEEE Computer Society. doi:10.1109/CVPR.2016.319.

Zoph, B.; and Le, Q. V. 2017. 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.

Zoph, B.; Vasudevan, V.; Shlens, J.; and Le, Q. V. 2018. Learning Transferable Architectures for Scalable Image Recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 8697–8710. IEEE Computer Society. doi:10.1109/CVPR.2018.00907.