Evolutionary Multi-objective Architecture Search Framework: Application to COVID-19 3D CT Classification

Xin He[1], Guohao Ying[2], Jiyong Zhang†[3], and Xiaowen Chu†[41⋆]

[1] Hong Kong Baptist University, Hong Kong, China
[2] University of Southern California, CA, USA
[3] Hangzhou Dianzi University, Hang Zhou, China
[4] The Hong Kong University of Science and Technology (Guangzhou), China

Abstract. The COVID-19 pandemic has threatened global health. Many studies have applied deep convolutional neural networks (CNN) to recognize COVID-19 based on chest 3D computed tomography (CT). Recent works show that no model generalizes well across CT datasets from different countries, and manually designing models for specific datasets requires expertise; thus, neural architecture search (NAS) that aims to search models automatically has become an attractive solution. To reduce the search cost on large 3D CT datasets, most NAS-based works use the weight-sharing (WS) strategy to make all models share weights within a supernet; however, WS inevitably incurs search instability, leading to inaccurate model estimation. In this work, we propose an efficient **E**volutionary **M**ulti-objective **AR**chitecture **S**earch (**EMARS**) framework. We propose a new objective, namely **potential**, which can help exploit promising models to indirectly reduce the number of models involved in weights training, thus alleviating search instability. We demonstrate that under objectives of accuracy and potential, EMARS can balance exploitation and exploration, *i.e.,* reducing search time and finding better models. Our searched models are small and perform better than prior works on three public COVID-19 3D CT datasets.

Keywords: COVID-19 · Neural Architecture Search (NAS) · Weight-sharing · Evolutionary Algorithm (EA) · 3D Computed Tomograph (CT)

1 Introduction

The rapid spread of *coronavirus disease 2019* (COVID-19) pandemic has threatened global health. Isolating infected patients is an effective way to block the transmission of the virus. Thus, fast and accurate methods to detect infected patients are crucial. Chest CT is relatively easy to perform and has been proved an important complement to nucleic acid test [7]. However, there is a serious lack of radiologists during the pandemic. Many researchers have applied deep

⋆ † Corresponding authors (xwchu@ust.hk; jzhang@hdu.edu.cn).

learning (DL) techniques to assist CT diagnosis. For COVID-19 3D CT classification, there are two mainstream CNN-based methods: 1) multiview-based methods [15, 22] use 2D CNN to extract features for each 2D CT slice and then fuse these features to make predictions; and 2) voxel-based methods [8, 32] feed 3D CNNs with 3D CT scans to make full use of the geometric information. He *et al.* [9] benchmark a series of hand-crafted 2D and 3D CNNs and demonstrate that 3D CNNs generally outperform 2D CNNs.

Some recent works [8, 11] benchmark multiple COVID-19 datasets from different countries and find that no model can maintain absolute advantages on different datasets. However, since it is difficult to design models manually for specific datasets, the neural architecture search (NAS) [6, 10] has become an attractive solution to discover superior models without human assistance. Reinforcement learning [21, 33], gradient descent (GD) [18], and evolutionary algorithm (EA) [23,30] are three mainstream NAS methods. The comparative results of a recent survey [10] show that the EA-based NAS can discover better networks than other types of NAS methods. However, the better performance of EA-based NAS is at the cost of more computing resources because they need to retrain all searched models to compare their performance, *e.g.,* AmoebaNet [23] took 3,150 GPU days to search. Thanks to the weight-sharing method [21, 29], any model can be evaluated without retraining, and Yang et al. [30] reduced the search time of the EA-based NAS to 0.4 GPU days. NAS was originally proposed for large-scale 2D image tasks. Although some works [8, 9] have extended NAS to search 3D models for COVID-19 3D datasets, they suffered from the search instability (analyzed in Sec. 3.1) incurred by weight-sharing, which leads to fluctuation in the search process and even worse results than random search in some cases. In this work, we propose an efficient **E**volutionary **M**ulti-objective **AR**chitecture **S**earch framework, dubbed as **EMARS**. We summarize our contributions below.

1. We propose a new objective, *i.e., potential*, which can help exploit promising models and indirectly reduce the number of models involved in weights training, thereby alleviating search instability.
2. We demonstrate that compared to conventional objective settings (*e.g.,* only considering accuracy), EMARS that aims at accuracy, potential, and small size objectives can trade-off between exploitation and exploration, reducing search time by 22% on average and discovering better models.
3. Our searched models are small in size and outperform prior works [8, 9] on three public datasets: CC-CCII [31], MosMed [20], and Covid-CTset [22].

2 Preliminaries

In this section, we describe the common basis of weight-sharing neural architecture search (NAS) [29]. NAS is formulated as a bi-level optimization problem:

$$\begin{aligned} &\min_{\alpha} L_{\text{val}}\ (w^*, \alpha) \\ &\text{s.t.}\quad w^* = \operatorname{argmin}_{w} L_{\text{train}}\ (w, \alpha) \end{aligned} \tag{1}$$

where L_{train} and L_{val} indicate the training and validation loss; w and α indicate the weights and architecture of a candidate model. The early NAS methods [23, 33] search and evaluate the networks by retraining them from scratch, resulting in huge computational cost. To reduce the burden, the weight-sharing strategy [29] was proposed, in which the SuperNet $\mathcal{N}$ contains all possible architectures (subnets) and its weights $\mathcal{W}$ are shared among these subnets. The architecture and weights of each subnet are denoted by $\mathcal{N}(\alpha)$ and $\mathcal{W}(\alpha)$, respectively, where α is the subnet architecture, encoded by one-hot sequences (described in Sec. 3.3). The loss of a subnet is expressed as $L(\alpha) = L(\mathcal{N}(\alpha), \mathcal{W}(\alpha), X, Y)$, where L, X, Y indicate the loss function, input data, and target, respectively, and the gradient of subnet weights is $\nabla_{\mathcal{W}(\alpha)} = \frac{\partial L(\alpha)}{\partial \mathcal{W}}$. Then gradients of SuperNet weights $\mathcal{W}$ can be calculated as the average gradient of all subnets, *i.e.,* $\nabla_{\mathcal{W}} = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\mathcal{W}(\alpha_i)} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial L(\alpha_i)}{\partial \mathcal{W}}$, where N is the total number of subnets. Obviously, it is not practical to use all subnets to update SuperNet weights at each time. Therefore, we use a mini-batch of subnets for training, detailed as Eq. 2

$$\nabla_{\mathcal{W}} \approx \frac{1}{M}\sum_{i=1}^{M} \nabla_{\mathcal{W}(\alpha_i)} \tag{2}$$

where M is the number of subnets sampled in a mini-batch and $M << N$. In our experiments, we find that $M = 1$ works just fine, *i.e.,* we can update $\mathcal{W}$ using the gradient from any single sampled subnets for each training batch.

3 Methodology

3.1 Potential objective: Alleviating Search Instability

By *instability*, we mean that the same subnet can produce a completely different performance at different times of the search process. The instability is caused by the weight-sharing strategy because the weights of all subnets are coupled, then an update of any subnet's weights is bound to affect (usually negatively) other subnets. Therefore, the performance of a subnet at a specific time does not necessarily represent its real performance but instead misleads the direction of evolutionary search (described in Sec. 3.2). To mitigate the search instability caused by weight-sharing, a natural idea is to reduce the number of models involved in weights training (*i.e.,* Eq. 2). For this reason, some works [2, 13] directly reduce the number of models by progressively shrinking the search space based on the model performance, but this may eliminate promising models in the early stage of the search. To avoid this problem, we take an indirect approach in which we keep exploring various models in the early stage of the search and then spend more effort on training those promising models in the later stage of the search. In this way, we can indirectly reduce the number of models involved in weights training without deliberately reducing the search space. However, how do we determine whether a model is promising or not?

Here, we propose a new objective, namely *potential*, to help find promising models. Specifically, for each sampled model, we maintain and update its historical performances $Z = (E, F)$, where $E = [e_1, ..., e_m]^T$ is a column vector recording the epochs when the model is sampled, $F = [f_1, ..., f_m]^T$ is a column vector recording the corresponding validation accuracy. Note that, Z is dynamically updated with the search process, so the size of Z (*i.e.,* m) varies for models. The potential $\mathcal{P}$ of a model is calculated by ordinary least squares (OLS):

$$\mathcal{P} = (E^T E)^{-1} E^T F \tag{3}$$

To some extent, E can also reflect how promising a model is, *e.g.,* if E is densely distributed, it means this model outperforms other models in multiple rounds of search and hence wins more chances to be sampled. However, considering only E will exacerbate the Matthew effect, and the search may get trapped in a local optimum. Our proposed potential solves this problem by considering the coupling relation between sampling frequency E and validation accuracy F, *i.e.,* the growing trend of accuracy rather than the accuracy at a specific time. The larger the $\mathcal{P}$ value, the more promising the model is.

3.2 Evolutionary Search

The search algorithm (see Supplement Alg. 1) starts with a warm-up stage, followed by the evolutionary search stage. In the warm-up, the SuperNet is trained by uniformly sampling subnets, thus all candidate operations are trained equally. After the warm-up, top-P best-performing subnets form the initial population, *i.e.,* $\mathcal{A}^{(0)}$, and will be evolved for multiple generations. Each generation comprises two sequential processes: 1) *weights training*, where each individual (*i.e.,* subnet) is selected from the population and trained based on Eq 2; and 2) *architecture search*, comprising selection, crossover, and mutation (see Fig. 1).

Selection. After weights training, we record multiple objectives for all individuals in the population. We adopt NSGA-II [4] method to select Pareto-front individuals under the recorded objectives from the population. We compare different combinations of these objectives in Sec. 4.2 and find that searching with potential and accuracy can discover better models with less cost.

Crossover&Mutation. The selection produces K Pareto-front individuals, based on which we further generate $P - K$ new individuals. Each new individual is generated by randomly sampling from the SuperNet or performing crossover and mutation (CM) with certain probabilities. The basic unit of CM is the one-hot sequence, representing the candidate operation (see Fig. 1).

Exploitation&Exploration. Fig. 1 (lower-right) shows an example of two important issues in the evolutionary algorithm (EA) based search: *exploration* and *exploitation*. Exploitation prefers the current optimal solution, which reduces search cost but may lead to a local optimum; exploration is more likely to find the optimal solution but consumes more resources. The common opinion about EA is that the steps of crossover and mutation determine the exploration, and exploitation is done by selection. However, our experiments in Sec. 4.2 show that

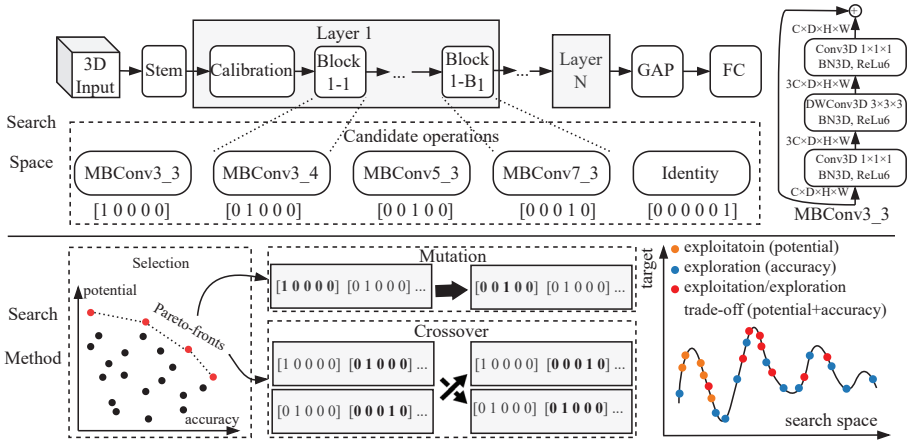

Fig. 1. Overview of search space and search method. Upper-right: MBConv3_3, where C, D, H, W indicate channels, depth, height, and width. Lower-right: An example of exploitation and exploration under different objectives. (best viewed in color)

setting different objectives in the selection step can also control the evolution direction. Specifically, accuracy and potential will make the evolution process towards exploration and exploitation, respectively, while combining accuracy and potential can balance exploration and exploitation.

3.3 Search Space

SuperNet. The search space is represented by a SuperNet $\mathcal{N}$, containing all possible subnets. SuperNet comprises two parts: 1) the searchable part, *i.e.,* $N = 6$ layers; 2) the fixed part, *i.e.,* stem block, global average pooling [17], and a fully connected layer. The stem block is a standard $3 \times 3 \times 3$ 3D convolution followed by a 3D batch normalization and a ReLu6 activation function [12].

Layer. The i-th layer comprises a calibration block and B_i searchable blocks. The calibration block is a 3D $1 \times 1 \times 1$ point-wise convolution to solve the problem of feature dimension mismatch; thus, all subsequent blocks have a stride of 1. The number of searchable blocks and the stride of calibration block in six layers are [4,4,4,4,4,1] and [2,2,2,1,2,1], respectively. The output channels of the stem block and six layers are 32 and [24,40,80,96,192,320], respectively.

Block. Each searchable block is a candidate operation, encoded by a one-hot sequence. We adopt eight candidate operations, including a *skip-connection* operation and seven mobile inverted bottleneck convolutions [24], denoted by *MBConvk_e*, where $k\_e \in \{3\_3, 3\_4, 3\_6, 5\_3, 5\_4, 7\_3, 7\_4\}$, k is the kernel size of the intermediate depth-wise convolution (DWConv), and e is the expansion ratio between the input channel and inner channel of MBConv.

4 Experiments

4.1 Implementation Details

Datasets. For a fair comparison, we apply the same three datasets as prior works [8,9]. CC-CCII [31] has 3,993 CT scans of three classes: novel coronavirus pneumonia (NCP), common pneumonia (CP), and normal case; MosMed [20] has 1,110 scans of NCP and normal classes; Covid-CTset [22] has 526 scans of NCP and normal classes. More details of datasets can be referred to supplement.

Search stage. We use four Nvidia V100 GPUs to search for 100 epochs, where the warm-up stage has 10 epochs. During each search epoch, a population of models are equally trained on the training set and evaluated on the validation set. The population size is 20, where 10 Pareto-front models are selected from the population using NSGA-II [4] under multiple objectives (*e.g.,* validation accuracy, potential, and model size), and 10 new models are generated by crossover and mutation with the probabilities of 0.3 and 0.2. To improve search efficiency, we set the input size ($width \times height \times depth$) to $64 \times 64 \times 16$. We use Adam optimizer [14] with a weight decay of 3e-4 and an initial learning rate of 0.001.

Retraining stage. After the search stage, we combine the training and validation set and retrain the Pareto-front models on the combined set for 200 epochs. We use the same Adam settings as the search stage. The 3D input sizes of CC-CCII, MosMed, and Covid-CTset datasets are $128 \times 128 \times 32$, $256 \times 256 \times 40$, and $512 \times 512 \times 32$, respectively. Our framework is based on NNI [19] and available at: `https://github.com/marsggbo/MICCAI2022-EMARS`.

4.2 Results and Analysis

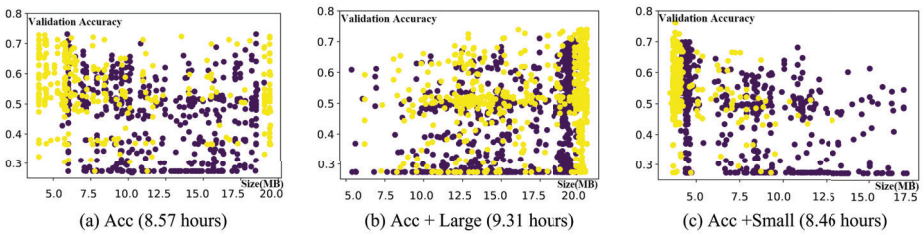

Fig. 2. The model size-aware search results. X and Y axes indicate model size and validation accuracy (Acc). The purple and yellow points indicate the sampled models in the first and last half of the search stage, respectively. (best viewed in color)

Model Size-aware Search. Fig. 2 presents model size-aware search results on CC-CCII dataset. Fig. 2 (a) shows that searching under only validation accuracy (Acc) will explore both extremes of model size, but with no performance gain, while Fig. 2 (b)&(c) show that additional consideration of model size on

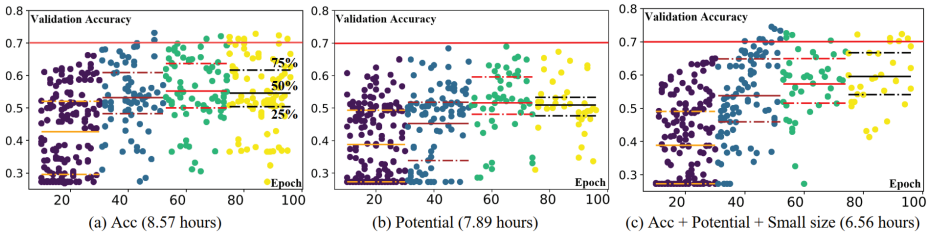

Fig. 3. The potential (P) aware search results. Different colored points indicate the models sampled in different epoch periods. The solid and dashed lines in each period indicate the average and 25/75 percentile accuracy, respectively. (best viewed in color)

top of Acc helps find better models in the later stage, indicating multi-objective can facilitate the search process. Besides, compared to Fig. 2 (b), searching under Acc and small model size in Fig. 2 (c) can not only reduce search time from 9.31 hours to 8.46 hours but also discover competitive models.

Potential-aware Search. We further build three experiments on the CC-CCII dataset to validate *potential* objective. Each sub-figure of Fig. 3 divides the search process into four periods based on the search epoch. Each period is presented with different colors and marked with the accuracy of 25/50/75th percentiles. Fig. 3 (a) shows that searching under Acc tends to *explore* more models, regardless of whether the model performance is good or bad, leading to wasting time on those unpromising models (lower-right points). On the contrary, in Fig. 3 (b), the difference between the 25th and 75th percentiles and the number of sampled models are gradually reduced with the search process, which implies that potential will guide the evolution process in the later stage to *exploit* promising models already discovered. Although it reduces search time, it has lower Acc due to being trapped in local optima in the early stage. Fig. 3 (c) shows that searching under potential, Acc, and small size can reduce the search time by 19% on average and balance exploitation and exploration. Specifically, the first two periods are dominated by exploration, as a wide accuracy range of models is explored, and we can find models with an accuracy of more than 0.7 faster in the second period. On the other hand, the last two periods focus more on exploitation, as the number of unpromising models is significantly reduced, and the accuracy of 25/50/75th percentiles is improved steadily.

Comparison with Prior Works. Table. 1 compares our searched models with prior works based on four widely used metrics: accuracy, precision, sensitivity, and f1 score. Precision and sensitivity are a pair of negatively correlated metrics, so they cannot fully describe model performance. F1 score is the harmonic mean of the precision and sensitivity; thus, it is a better metric. As can be seen, our models searched under APS (accuracy, potential, and small model size) objectives have small sizes and outperform all prior hand-crafted and NAS-based models on three datasets in terms of accuracy, precision, and f1 score. Besides, MosMed is an imbalanced dataset, and we can find that the models (*e.g.,* CovidNet3D-S/L and EMARS-A) searched without potential are

overfitted on positive class (*i.e.,* NCP), as they have extremely high sensitivity but low precision. On the contrary, EMARS-P and EMARS-APS are searched with potential objective, balancing precision and sensitivity well and achieving higher accuracy and f1 scores. More results can be referred to the Supplement.

Table 1. Results on CC-CCII [31], MosMed [20], and Covid-CTset [22] datasets. A, P, and S in our model name indicate accuracy, potential, and small model size, *e.g.,* EMARS-A indicates the model searched under the accuracy objective.

Dataset	Model	Size (MB)	Type	Accuracy	Precision	Sensitivity	F1
CC-CCII [China] [31]	ResNet3D101 [26]	325.21	Manual	85.54	89.62	77.15	82.92
	DenseNet3D121 [5]	43.06		87.02	88.97	82.78	85.76
	MC3_18 [26]	43.84		86.16	87.11	82.78	84.89
	COVID-AL [28]	-		86.60	-	-	-
	VGG16-Ensemble [16]	-		88.12	84.04	89.19	86.54
	CovidNet3D-S [8]	11.48	Auto	88.55	88.78	**91.72**	90.23
	CovidNet3D-L [8]	53.26		88.69	90.48	88.08	89.26
	MNas3DNet [9]	22.91		87.14	88.44	86.09	87.25
	EMARS-A	5.93		**89.67**	89.26	89.22	89.23
	EMARS-P	5.63		88.78	88.81	88.22	88.51
	EMARS-APS	3.38		89.61	**91.48**	89.97	**90.72**
MosMed [Russia] [20]	ResNet3D101 [26]	325.21	Manual	81.82	81.31	97.25	88.57
	DenseNet3D121 [5]	43.06		79.55	84.23	92.16	88.01
	MC3_18 [26]	43.84		80.4	79.43	98.43	87.92
	DeCoVNet [27]	-		82.43	-	-	-
	CovidNet3D-S [8]	12.48	Auto	81.17	78.82	99.22	87.85
	CovidNet3D-L [8]	60.39		82.29	79.50	98.82	88.11
	EMARS-A	2.89		80.98	77.91	**99.61**	87.44
	EMARS-P	18.22		84.34	**93.56**	85.49	89.34
	EMARS-APS	10.69		**88.09**	93.52	90.59	**92.03**
Covid-CTset [Iran] [22]	ResNet3D101 [26]	325.21	Manual	93.87	92.34	95.54	93.92
	DenseNet3D121 [5]	43.06		91.91	92.57	92.57	92.57
	MC3_18 [26]	43.84		92.57	90.95	94.55	92.72
	CovCTx [3]	-		96.37	-	97.00	-
	Vit-32×32 [25]	-		95.36	-	83.00	-
	CovidNet3D-S [8]	8.36	Auto	94.27	92.68	90.48	91.57
	CovidNet3D-L [8]	62.82		96.88	97.50	92.86	95.12
	AutoGluon model [1]	93.00		89.00	90.00	88.00	88.00
	EMARS-A	8.36		95.16	95.77	95.16	95.46
	EMARS-P	14.41		92.87	92.73	92.74	92.74
	EMARS-APS	9.95		**97.66**	**97.61**	**97.58**	**97.59**

5 Conclusion and Future Work

In this work, we introduce an EA-based neural architecture search (EMARS) framework, which can efficiently discover superior 3D models under multiple ob-

jectives for COVID-19 3D CT classification. We demonstrate that our proposed objective, *i.e., potential*, can effectively alleviate the search instability and help exploit promising models. The models searched by EMARS under accuracy and potential objectives have small sizes and outperform the previous work on three public datasets. We believe our framework can also be extended to other types of datasets and tasks (*e.g.,* segmentation), which is also our future work.

6 Acknowledge

This work was supported in part by Hong Kong Research Matching Grant RMGS2019_1_23, the Zhejiang Province Nature Science Foundation of China under Grant LZ22F020003, and the HDU-CECDATA Joint Research Center of Big Data Technologies under Grant KYH063120009.

References

1. Anwar, T.: Covid19 diagnosis using automl from 3d ct scans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 503–507 (2021)
2. Chen, M., Fu, J., Ling, H.: One-shot neural ensemble architecture search by diversity-guided search space shrinking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16530–16539 (2021)
3. Chetoui, M., Akhloufi, M.A.: Efficient deep neural network for an automated detection of covid-19 using ct images. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 1769–1774. IEEE (2021)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
5. Diba, A., Fayyaz, M., Sharma, V., Karami, A.H., Arzani, M.M., Yousefzadeh, R., Van Gool, L.: Temporal 3d convnets: New architecture and transfer learning for video classification. arXiv preprint arXiv:1711.08200 (2017)
6. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. arXiv preprint arXiv:1808.05377 (2018)
7. Fu, Z., Tang, N., Chen, Y., Ma, L., Wei, Y., Lu, Y., Ye, K., Liu, H., Tang, F., Huang, G., et al.: Ct features of covid-19 patients with two consecutive negative rt-pcr tests after treatment. Scientific Reports **10**(1), 1–6 (2020)
8. He, X., Wang, S., Chu, X., Shi, S., Tang, J., Liu, X., Yan, C., Zhang, J., Ding, G.: 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 pp. 4821–4829 (May 2021)
9. He, X., Wang, S., Shi, S., Chu, X., Tang, J., Liu, X., Yan, C., Zhang, J., Ding, G.: Benchmarking deep learning models and automated model design for covid-19 detection with chest ct scans. medRxiv (2020)
10. He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems **212**, 106622 (2021)
11. Horry, M.J., Chakraborty, S., Pradhan, B., Fallahpoor, M., Chegeni, H., Paul, M.: Factors determining generalization in deep learning models for scoring covid-ct images. Mathematical Biosciences and Engineering **18**(6), 9264–9293 (2021)

12. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Mobilenets, H.A.: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
13. Hu, Y., Liang, Y., Guo, Z., Wan, R., Zhang, X., Wei, Y., Gu, Q., Sun, J.: Angle-based search space shrinking for neural architecture search. In: European Conference on Computer Vision. pp. 119–134. Springer (2020)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR (2015)
15. Li, L., Qin, L., Xu, Z., Yin, Y., Wang, X., Kong, B., Bai, J., Lu, Y., Fang, Z., Song, Q., et al.: Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct. Radiology p. 200905 (2020)
16. Li, X., Tan, W., Liu, P., Zhou, Q., Yang, J.: Classification of covid-19 chest ct images based on ensemble deep learning. Journal of Healthcare Engineering **2021** (2021)
17. Lin, M., Chen, Q., Yan, S.: Network in network. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR (2014)
18. Liu, H., Simonyan, K., Yang, Y.: DARTS: differentiable architecture search. In: 7th International Conference on Learning Representations, ICLR (2019)
19. Microsoftware: Neural network intelligence (nni). `https://github.com/microsoft/nni/tree/v1.4` (2019)
20. Morozov, S., Andreychenko, A., Pavlov, N., Vladzymyrskyy, A., Ledikhova, N., Gombolevskiy, V., Blokhin, I., Gelezhe, P., Gonchar, A., Chernina, V., Babkin, V.: Mosmeddata: Chest ct scans with covid-19 related findings. medRxiv (2020)
21. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 4092–4101. PMLR (2018)
22. Rahimzadeh, M., Attar, A., Sakhaei, S.M.: A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. medRxiv (2020)
23. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: The Thirty-Third AAAI Conference on Artificial Intelligence. pp. 4780–4789. AAAI Press (2019)
24. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4510–4520. IEEE Computer Society (2018)
25. Than, J.C., Thon, P.L., Rijal, O.M., Kassim, R.M., Yunus, A., Noor, N.M., Then, P.: Preliminary study on patch sizes in vision transformers (vit) for covid-19 and diseased lungs classification. In: 2021 IEEE National Biomedical Engineering Conference (NBEC). pp. 146–150. IEEE (2021)
26. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 6450–6459. IEEE Computer Society (2018)
27. Wang, X., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., Zheng, C.: A weakly-supervised framework for covid-19 classification and lesion localization from chest ct. IEEE transactions on medical imaging **39**(8), 2615–2625 (2020)

28. Wu, X., Chen, C., Zhong, M., Wang, J., Shi, J.: Covid-al: The diagnosis of covid-19 with deep active learning. Medical Image Analysis **68**, 101913 (2021)
29. Xie, L., Chen, X., Bi, K., Wei, L., Xu, Y., Wang, L., Chen, Z., Xiao, A., Chang, J., Zhang, X., et al.: Weight-sharing neural architecture search: A battle to shrink the optimization gap. ACM Computing Surveys (CSUR) **54**(9), 1–37 (2021)
30. Yang, Z., Wang, Y., Chen, X., Shi, B., Xu, C., Xu, C., Tian, Q., Xu, C.: CARS: continuous evolution for efficient neural architecture search. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1826–1835. IEEE (2020)
31. Zhang, K., Liu, X., Shen, J., Li, Z., Sang, Y., Wu, X., Zha, Y., Liang, W., Wang, C., Wang, K., et al.: Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography. Cell (2020)
32. Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., Wang, X.: Deep learning-based detection for covid-19 from chest ct using weak label. MedRxiv (2020)
33. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. In: 5th International Conference on Learning Representations, ICLR 2017 (2017)