IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 9, 2021, accepted March 15, 2021, date of publication March 24, 2021, date of current version April 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068766

Differentiable Architecture Search Based on Coordinate Descent

PYUNGHWAN AHN[1], HYEONG GWON HONG[2], AND JUNMO KIM[1,2], (Member, IEEE)
[1]School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[2]Graduate School of AI, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Junmo Kim (junmo.kim@kaist.ac.kr)

This work was supported by the Center for Applied Research in Artificial Intelligence (CARAI) Grant through the Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) under Grant UD190031RD.

ABSTRACT Neural architecture search (NAS) is an automated method searching for the optimal network architecture by optimizing the combinations of edges and operations. For efficiency, recent differentiable architecture search methods adopt a one-shot network, containing all the candidate operations in each edge, instead of sampling and training individual architectures. However, a recent study doubts the effectiveness of differentiable methods by showing that random search can achieve comparable performance with differentiable methods using the same search cost. Therefore, there is a need to reduce the search cost even for previous differentiable methods. For more efficient differentiable architecture search, we propose a differentiable architecture search based on coordinate descent (DARTS-CD) that searches for optimal operation over only one sampled edge per training step. DARTS-CD is proposed based on the coordinate descent algorithm, which is an efficient learning method for resolving large-scale problems by updating only a subset of parameters. In DARTS-CD, one edge is randomly sampled, in which all the operations are performed, whereas only one operation is applied to the other edges. Weight update is also performed only at the sampled edge. By optimizing each edge separately, as in the coordinate descent that optimizes each coordinate individually, DARTS-CD converges much faster than DARTS while using the network architecture similar to that used for evaluation. We experimentally show that DARTS-CD performs comparably to the state-of-the-art efficient architecture search algorithms, with an extremely low search cost of 0.125 GPU days (1/12 of the search cost of DARTS) on CIFAR-10 and CIFAR-100. Furthermore, a warm-up regularization method is introduced to improve the exploration capability, which further enhances the performance.

INDEX TERMS Automatic machine learning (AutoML), differentiable architecture search (DARTS), neural architecture search (NAS).

I. INTRODUCTION

Over the past few years, deep neural networks have shown remarkable performance in many computer vision tasks such as object recognition [1]–[5], object detection [6]–[8], and semantic segmentation [9]–[11]. Researchers have attempted to design specialized architectures for each task. When a new task or a new dataset arises, human experts' trial-and-error search methods are time-consuming. To resolve this issue, recent studies have attempted to automate the process of architecture search.

Most recent neural architecture search (NAS) studies have focused on searching for a better performing architecture efficiently, without considerable computational burden. A majority of NAS works perform searches on cell structures (see FIGURE 1), building blocks of networks such as residual blocks for ResNet [4]. When the NAS problem is downscaled to the cell structure decision problem, it significantly reduces the search space. However, state-of-the-art methods based on reinforcement learning [12] or evolutionary algorithms [13] still require thousands of GPU days for the search. Thus, they are not considered an appropriate tool for finding the optimal architecture for a new task or a new dataset.

In the spirit of efficient architecture search, differentiable architecture search (DARTS) [14] was proposed to train the one-shot network [15], which contains all the candidate operations and their corresponding coefficients, called architecture parameters. The architecture parameters are also

The associate editor coordinating the review of this manuscript and approving it for publication was Corrado Mencar.

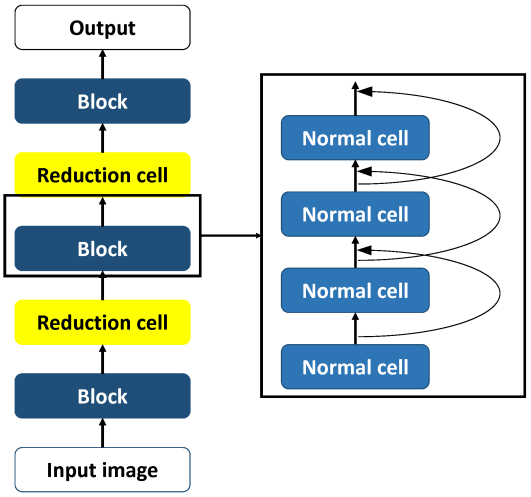


FIGURE 1. Network divided into repeated cells. Normal cells produce output of the same spatial dimension, while reduction cells downscale input (usually into half) along each spatial dimension. The number of repeated cells can be decided depending on the dataset's complexity and may be different during the search and evaluation stages.

trained with gradient descent and expected to evaluate the importance of the corresponding operation. After the search, the final network architecture is derived by choosing one operation per edge according to the architecture parameter values. Although a single one-shot network is trained instead of multiple candidate networks, DARTS runs in three orders of magnitude faster than NASNet [12] or AmoebaNet [13] and still shows comparable performance.

Although DARTS has been suggested as an efficient alternative to computationally heavy NAS methods, several studies have highlighted the inefficiency of the method. For example, Li and Talwalkar [16] showed that a random search can identify architectures with performances comparable to those that result from DARTS within the same time limit. Owing to the absence of search costs in random selection, multiple architectures are sampled and evaluated through early stopping during a single round of the DARTS search phase. This implies that DARTS does not effectively spend training time to identify the optimal architecture. To allow a differentiable architecture search to be more meaningful, the search time should be improved further.

In various optimization problems, the coordinate descent algorithm is adopted as an efficient approach compared with other methods [17]–[20]. Coordinate descent achieves rapid convergence by updating only the parameters of the selected coordinates. Accordingly, we propose differentiable architecture search based on coordinate descent (DARTS-CD) to leverage coordinate descent in training a one-shot network. In this framework, a sampled subset of the architecture parameters is updated in each training step. DARTS-CD is efficient in both memory and computation, showing better performance under a search cost of $1/12$ and a memory cost of $1/8$ compared with DARTS.

DARTS-CD is also an alternative method to bridge the gap between the derived network performance and one-shot network performance. There is inconsistency between the derived network and one-shot network in DARTS [21]. This is because the search progress of DARTS is dominated by the loss function for one-shot network training. In DARTS-CD, candidate operations are mingled for only selected coordinate; thus, the training loss in the search stage is more relevant to the performance of the derived network.

DARTS-CD limits exploration over the search space owing to the constrained update directions of the architecture parameters. To compensate for this, a regularization method called *warm-up* is proposed, which inverts the signs of all the architecture parameters to provide fair learning opportunities to all candidate operations. This novel regularization method can enhance the exploration ability of DARTS-CD resulting in better performance than the standalone version.

The primary contributions of this study can be summarized as follows:

- DARTS-CD is the first framework to apply the coordinate descent algorithm to the training of architecture parameters in a one-shot network, which is efficient in both memory and computation.
- In DARTS-CD, the training objective optimizes the one-shot network and derived network, which alleviates the inconsistency issue highlighted in [21].
- To compensate for the reduced exploration caused by the nature of coordinate descent, we propose a simple regularization method that manually explores the search space during the early stages of the search.
- We demonstrate that DARTS-CD shows comparable performance to DARTS with a search time and memory consumption of approximately $1/12$ and $1/8$, respectively, on benchmark datasets CIFAR-10 and CIFAR-100.

II. RELATED WORKS

Since the innovation of CNNs, led by AlexNet [1], several experts have manually designed variants of CNN architectures. Following Krizhevsky *et al.* [1], architectures with smaller filter sizes [2], multi-path structures [3], or extremely deep networks [4] have been proposed. In recent years, some efforts have been made, such as widening the network instead of increasing the depth [22] or densely connecting layers [5] to ease the gradient flow.

Automatic search for neural network architectures has attracted attention in recent years. There are two main streams in the development of early NAS methods, including reinforcement learning-based methods [12], [23] and evolution-based methods [13]. In NASNet [12], several architectures are sampled from a controller and trained to a certain extent to estimate performance. The controller then judges how the architecture should be changed to maximize the estimated performance. Because the process of training multiple models of different architectures is computationally heavy,

Pham *et al.* [23] proposed to share model weights among the sampled architectures. Some other methods, as suggested in [13], share the sampling and evaluation process of NASNet; however, they use evolutionary algorithms as meta controller.

Recently proposed DARTS [14] is an efficient and effective algorithm, that takes only 1.5–4 GPU days for the search stage. DARTS is a one-shot architecture search method that includes all candidate paths in a single overparameterized network called a one-shot network, as in Bender *et al.* [15]. In a gradient-based approach such as DARTS [14], architecture parameters are trained using a separate data split from that used for operation parameters in the one-shot network, and used to choose operations that can obtain higher validation accuracy. The gradient-based method is extremely efficient compared to previous search methods by Zoph *et al.* [12] and Real *et al.* [13]. The operation parameters of all the candidate architectures can be obtained by training a single one-shot network.

After the introduction of DARTS, several attempts have been made to improve the differentiable search. [21] stated that the information flow in the search stage differs from that in the evaluation stage in DARTS. This problem is verified by a dramatic performance drop when evaluating the derived architecture with the operation parameters transferred from the trained one-shot network in DARTS. In [21], stochastic NAS was suggested to alleviate this gap by applying Gumbel–Softmax to the architecture parameters to optimize discrete random variables. Another method named P-DARTS [24] was proposed to mitigate the depth gap problem in DARTS. The one-shot network is shallower than the evaluation network in DARTS, making it less reasonable to determine architectures based on the architecture parameters trained in a one-shot network. Thus, Chen *et al.* [24] suggested building the network gradually deeper in the search stage while addressing the memory issue by reducing the number of candidate operations whenever the depth is increased.

ProxylessNAS [25] suggested sampling a single stochastic path in a one-shot network, explicitly using architecture parameters such as the sampling probability. This also bridges the gap in the cell structures between the search and evaluation stages of DARTS. In addition, it allows direct searching for networks of substantial depth on a large scale dataset such as ImageNet, thus being proxyless. Because only a single operation is computed at each edge, the memory issue is significantly alleviated during search.

Although variants of DARTS have been suggested to resolve various shortcomings of DARTS, most of them still have a gap in network behavior because of the difference in path activation in the search and evaluation stages. This gap originates from the difference in path activation, which requires further investigation. In this study, we propose to alleviate this gap by introducing coordinate descent into optimization; thus, the proposed method has a path activation scheme that is different from any other method, discussed in Section VI-A.

III. METHOD

A. PRELIMINARY: DARTS

Before introducing the proposed method, we summarize how the differentiable one-shot architecture search works and define the notations that will appear in the following subsections. Most of this subsection refers to the explanation of Liu *et al.* [14]. In a differentiable one-shot architecture search, all candidate operations in every edge are included in a single overparameterized network, called one-shot network. Each operation has a corresponding architecture parameter in this network, which increases as the operation becomes more likely to influence the network performance. After the search stage ends, DARTS determines the optimal architecture by selecting one operation among candidate operations for each edge with the highest architecture parameter value.

In DARTS, the network is represented as repeated cell structure, as in other recent NAS methods (FIGURE 1). Based on state-of-the-art convolutional neural network (CNN) architectures [4], [5], the network is divided into multiple stages, each of which consists of repeated cell structures, called normal cells. Between the stages, reduction cells are located to downsample feature maps along the spatial dimensions. Both normal and reduction cells are in the form of a directed acyclic graph (DAG) with N_n nodes.

Inside a cell, each edge connects a pair of nodes (i, j), where $0 \leq i < j \leq N_n - 1$, and performs specific operations on the node i to generate features for the node j. The output of an edge is a weighted sum of features processed by each candidate operation:

$$f_j(x_i) = \Sigma_{o \in O} \frac{\exp \alpha_o^{(i,j)}}{\Sigma_{o' \in O} \exp \alpha_{o'}^{(i,j)}} * o(x_i) \tag{1}$$

where x_i denotes the i-th node's value, o is a candidate operation in O, and α denotes the architecture parameter for each operation. Each intermediate node represents the sum of the results from all edges connected to the preceding nodes. The output node of a cell is the concatenation of all the intermediate nodes.

The architecture parameters (α) in the same edge are bound with a softmax function to compute importance weights for each operation, making DARTS fully differentiable. This is the key to continuous relaxation of the discrete search problem. Accordingly, all the one-shot network parameters, including the architecture parameters, can be trained by gradient descent. The network parameters are trained through bi-level optimization, using training data for operation parameters and validation data for architecture parameters. Thus, the network architecture with better generalization capability can be determined using separate data for each set of parameters. After the search stage, the cell structure is derived by pruning each edge to have only the max-α operation.

Then, the network for evaluation is constructed by repeating the derived cell structure, which is trained from scratch to evaluate the final performance.

B. PROPOSED METHOD

In this section, we introduce the proposed method and describe how it improves DARTS. The max-α method is explained first, followed by the DARTS-CD framework. DARTS-CD reduces the search cost by forwarding only one operation, instead of all the candidate operations, in most edges. Thus, both computation and memory costs are reduced by approximately a factor of N_o, which is the number of candidate operations. The search direction of DARTS-CD is constrained compared with that of DARTS, although being better than that of the max-α method. To overcome this limitation, we additionally propose a regularization technique.

1) MAX-α

Here, we propose to sample one operation per edge at every training step. At each edge, the operation with the maximum architecture parameter value (α) is sampled so that the network's active path is the same as the model derived at that moment. Then, the operation parameters are trained in this network, preventing the aforementioned performance gap.

Although this method significantly alleviates the performance inconsistency problem, the search direction is biased toward initially selected operations because of the absence of continuous relaxation. If only one operation is trained at each training step, the architecture parameter of that operation is more likely to increase, limiting the learning opportunity for other operations. We will experimentally verify this in Section V-C.

2) DARTS-CD

DARTS-CD employs coordinate descent to use both differentiable search and sampling-based search to determine the optimal architectures efficiently. In DARTS-CD, the parameter update is carried out at only one edge in every training step. As shown in FIGURE 2 (left), DARTS-CD randomly samples one edge (red arrow) and trains all the operation parameters at that edge. Meanwhile, the other edges (black arrow) process the input signal through the max-α operation (blue arrow). The following equation formulates this procedure:

$$f_j(x_i) = \begin{cases} \Sigma_{o\in O} \dfrac{\exp \alpha_o^{(i,j)}}{\Sigma_{o'\in O} \exp \alpha_{o'}^{(i,j)}} * o(x_i) & \text{if } (i,j) = (k,l) \\ o_m(x_i) & \text{otherwise} \end{cases} \quad (2)$$

where (k, l) is sampled from a uniform distribution over the set of all the edges, and o_m is the operation with the maximum α. The algorithm is described in detail in Algorithm 1.

DARTS-CD has the advantages of both differentiable and sampling-based search algorithms. It does not exhibit performance inconsistency, which is a critical problem of DARTS,

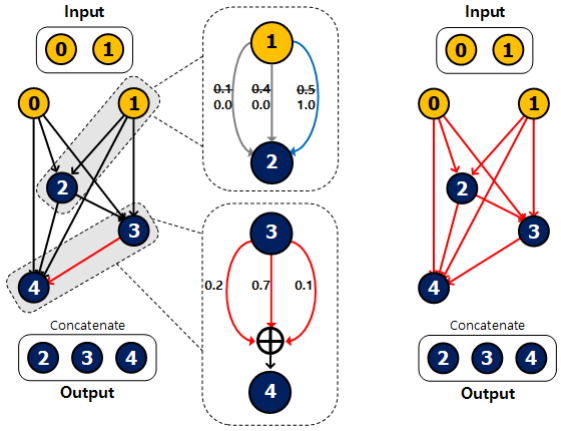


FIGURE 2. Schematic comparison between DARTS-CD (ours, left) and DARTS (right). An example with 6 nodes is presented, including 2 input nodes (yellow circles), 3 intermediate nodes (dark blue circles), and 1 output node. Edges represented by red arrows compute a weighted sum of all candidate operations while those represented as black arrows indicate only one operation (light blue arrow). Only one edge is stochastically sampled as a red edge in DARTS-CD (left), whereas in DARTS (right), all edges are red.

Algorithm 1: Forward Algorithm of the Cell in DARTS-CD

Input: Two input nodes x_0 and x_1, the number of intermediate nodes N, the set of candidate operation sets for each edge $\{O_{i,j},\ 0 \le i < j \le N+1\} \setminus \{O_{0,1}\}$, set of architecture parameters over all edges and operations $\{\alpha^o_{i,j},\ 0 \le i < j \le N+1, o \in O_{i,j}\} \setminus \{\alpha^o_{0,1},\ o \in O_{0,1}\}$, uniform distribution over all the edges U

Result: Output node y

sample $(k, l) \sim U$

allocate temporary zero tensors x_p $(2 \le p \le N+1)$

for $j := 2$ *to* $N+1$ **do**

  for $i := 0$ *to* $j-1$ **do**

    if $(i, j) = (k, l)$ **then**

      $x_j \leftarrow x_j + \sum_{o\in O_{i,j}} \frac{exp(\alpha^o_{i,j})}{\sum_{o'\in O_{i,j}} exp(\alpha^{o'}_{i,j})} o(x_i)$

    else

      $x_j \leftarrow x_j + (argmax_{o\in O_{i,j}}\ exp(\alpha^o_{i,j}))(x_i)$

    end if

  end

end

$y \leftarrow concat(x_2, x_4, \ldots, x_{N+1})$

as discussed in Section VI-B. Furthermore, the parameters of the max-α operation and the others are trained using stochastic continuous relaxation, facilitating changes of max-α operation, as discussed in Section V-C.

3) WARMUP REGULARIZATION

We suggest a regularization method to further improve the exploration capability of DARTS-CD. Sampling-based

methods, either in all of the edges or in most of them, limits the exploration of the operations that have lower architecture parameters at the beginning of training. To alleviate this problem, we propose to invert the sign of every architecture parameter once per epoch, for a certain period at the beginning of the search stage. In those epochs in which the signs are inverted, the edge (i, j) is computed as follows:

$$f_j(x_i) = \Sigma_{o\in O} \frac{\exp\left(-\alpha_o^{(i,j)}\right)}{\Sigma_{o'\in O} \exp\left(-\alpha_{o'}^{(i,j)}\right)} * o(x_i) \tag{3}$$

where all the notations follow equation 2. Based on this regularization, non-max-α operations have a better chance of obtaining higher weights.

IV. EXPERIMENTS

The training procedure of DARTS-CD consists of two separate stages following the DARTS pipeline [14]: the search and the evaluation stage.

In the **search stage**, DARTS-CD searches for the optimal cell structure. The most significant difference from DARTS is that DARTS-CD can perform search in a deeper network architecture with the same hardware resources. This is because the input is processed with only one operation at most of the edges, which significantly reduces memory requirements. In the following **evaluation stage**, the derived network is trained afresh, based on the cell structures determined in the search stage, and then the test accuracy is reported.

All the experiments were performed on CIFAR-10 and CIFAR-100, using a single NVIDIA Titan X (Pascal) GPU (with 12GB VRAM). A batch size of 64 was used for all experiments, which was the maximum size under the memory constraint.

A. SEARCH STAGE

1) SEARCH SPACE

Following recent studies including DARTS [14] and P-DARTS [24], we used a candidate operation set of seven operations: 1) 3×3 separable convolution, 2) 5×5 separable convolution, 3) 3×3 dilated separable convolution, 4) 5×5 dilated separable convolution, 5) 3×3 max pooling, 6) 3×3 average pooling, and 7) skip connection. For the actual implementation, an additional zero operation was included as the eighth operation.

2) DEEPER ONE-SHOT NETWORK

In DARTS, a one-shot network consists of eight cells: two normal cells in each of the three stages and two reduction cells between stages. The term 'stage' refers to the set of blocks with the same feature map size in the residual network. The network is trained by alternatively optimizing the operation parameters and architecture parameters (i.e., bi-level optimization). In DARTS-CD, the stochastic single-edge modification scheme uses a deeper one-shot network of 20 cells (six normal cells for each of the three stages and two reduction cells) despite the 12GB memory constraint of a single (Titan X) GPU. The Titan X GPU hardware used in this study has the lowest performance among the GPU hardware used in the previous studies. Both DARTS [14] and ENAS [23] used NVIDIA GTX 1080Ti GPU, whereas P-DARTS [24] and SNAS [21] used NVIDIA Tesla P100 GPU and NVIDIA Titan XP GPU, respectively.

3) EXPERIMENTAL SETTINGS

The one-shot network was trained for 25 epochs with a batch size of 64 using bi-level optimization (as in DARTS). Meanwhile, only the parameters in one randomly sampled edge were trained. The operation parameters were trained by the stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay of rate 3e-4. The learning rate was scheduled by the cosine annealing with initial and minimum learning rate of 0.025 and 0.001 respectively. The architecture parameters were trained using Adam optimizer with a learning rate of 0.025, betas of 0.5 and 0.999, and no weight decay. Weight decay regularization is removed to prevent the edges from being updated when they are not selected in the training step. Thus, DARTS-CD only focuses on modifying the architecture parameters for the selected edge. However, in the initial warmup setting, the weight decay rate was set to 1e-3, same as DARTS, to inhibit the architecture parameters from selecting the optimal cell too early. In addition, warmup regularization was applied for the first 10 epochs, inverting the sign of all the architecture parameters after every epoch.

B. EVALUATION STAGE

1) ARCHITECTURE DERIVATION

We used the derivation algorithm in DARTS and considered the operation with the maximum architecture parameter at each edge as the optimal choice. Thus, the cell structure was determined by pruning all operations other than the max-α operation at each edge. After derivation, the network was trained from scratch.

2) EXPERIMENTAL SETTINGS

The derived network was trained for 600 epochs. The settings other than the number of training epochs were kept the same as in the search stage.

V. RESULTS

A. EVALUATION

TABLE 1 presents a comparison of DARTS-CD with the state-of-the-art algorithms. DARTS-CD exhibits better performance than DARTS (see "DARTS (first order)" and "DARTS-CD-warmup CIFAR-10"), although the search cost is reduced to 1/12. DARTS-CD operates within only 3h on a single Titan X GPU, which allows a fast search of architectures for a new task. In addition, compared with other efficient methods, such as ENAS and SNAS, DARTS-CD shows comparable results with considerably smaller search cost.

TABLE 1. Comparison with state-of-the-art NAS methods on CIFAR-10 and CIFAR-100. "c.o." stands for cutout regularization [26].

Architecture	Test Err. (%)		Params (M)	Search Cost (GPU days)	Search Method
	C10	C100			
DenseNet-BC [5]	3.46	17.18	25.6	-	manual
AmoebaNet-A + c.o. [13]	3.34	-	3.2	3150	evolution
AmoebaNet-B + c.o. [13]	2.55	-	2.8	3150	evolution
NASNet-A + c.o. [12]	2.65	-	3.3	1800	RL
Hierarchical Evo [27]	3.75	-	15.7	300	evolution
PNAS [28]	3.41	-	3.2	225	SMBO
ProxylessNAS + c.o. [21]	2.08	-	5.7	4.0	gradient-based
DARTS (first order) + c.o. [14]	3.00	17.76	3.3	1.5	gradient-based
DARTS (second order) + c.o. [14]	2.76	17.54	3.3	4.0	gradient-based
SNAS + mild + c.o. [21]	2.98	-	2.9	1.5	gradient-based
SNAS + moderate + c.o. [21]	2.85	-	2.8	1.5	gradient-based
SNAS + aggressive + c.o. [21]	3.10	-	2.3	1.5	gradient-based
ENAS + c.o. [23]	2.89	-	4.6	0.5	RL
P-DARTS CIFAR-10 + c.o. [24]	2.50	16.55	3.4	0.3	gradient-based
P-DARTS CIFAR-100 + c.o. [11]	2.62	15.92	3.6	0.3	gradient-based
DARTS-CD (first order) + c.o. (ours)	2.92	17.61	3.44	0.125	gradient-based
DARTS-CD (first order) + warmup + c.o. (ours)	2.86	17.16	3.15	0.125	gradient-based

We also report the evaluation results of the architectures discovered on the CIFAR-100 dataset, on which only few previous methods exhibited performance improvement. The results indicate that DARTS-CD can effectively perform search on CIFAR-100 dataset, showing improved results over DARTS.

B. SEARCH BEHAVIOR

FIGURE 3 qualitatively demonstrates how the decision of cell structure changes during the search stage. The network was relatively shallow during the early period of training (See FIGURE 3 (b)–(d)). We attribute this phenomenon to the easier optimization of shallower networks. As training progresses, the network depth increases, which improves the network performance. (See FIGURE 3 (e) and (f)) The cell structure changes more rapidly in the first few epochs. Once the parameters are trained to a certain degree, it gradually changes for the rest of the training process. The structures determined by DARTS-CD, as shown in FIGURE 3 (b)–(f) appears to be similar.

We also conduct a quantitative analysis of the search progress in terms of parameter size and test accuracy of the models built with the cell structures, as shown in FIGURE 3. As described in TABLE 2, the test accuracy of the discovered network increases as the search progresses, thus demonstrating that DARTS-CD successfully searches for a better structure. There was no distinct pattern according to the change in the cell structure during the search stage in terms of the parameter size.

C. ARCHITECTURE PARAMETER DYNAMICS

This section analyzes how the architecture parameters change during the search stage of the proposed methods and presents some examples in FIGURE 4. All these methods are based on sampling. Briefly reiterating each method, **max-α** samples the operation with the maximum α at every edge, **DARTS-CD** samples one edge to be relaxed and use max-α operation for the others, and **DARTS-CD-warmup** refers to DARTS-CD with the sign of the architecture parameters inverted during the first few epochs.

We randomly selected three edges from any cell. We recorded the value of the architecture parameters after applying the softmax function to observe how they change during the search stage. We observed that in **max-α**, the architecture parameter rankings changed less often, and thus, the cell structures were determined early. In **DARTS-CD**, the ranking of architecture parameters changes slightly more often; however, the operation with the maximum α still rarely changes. This behavior suggests that the partial relaxation of the search problem, which is the change from max-α to DARTS-CD, is not sufficient for desired exploration capability. In **DARTS-CD-warmup**, the rankings of the architecture parameters change dynamically during the entire search stage, thereby allowing more exploration over the operations with lower α. Fair experience among the candidate operations helps the ranking to change even after the warmup stage. Accordingly, we conclude that the performance improvement of DARTS-CD-warmup compared with DARTS-CD originates from the increased exploration capability.

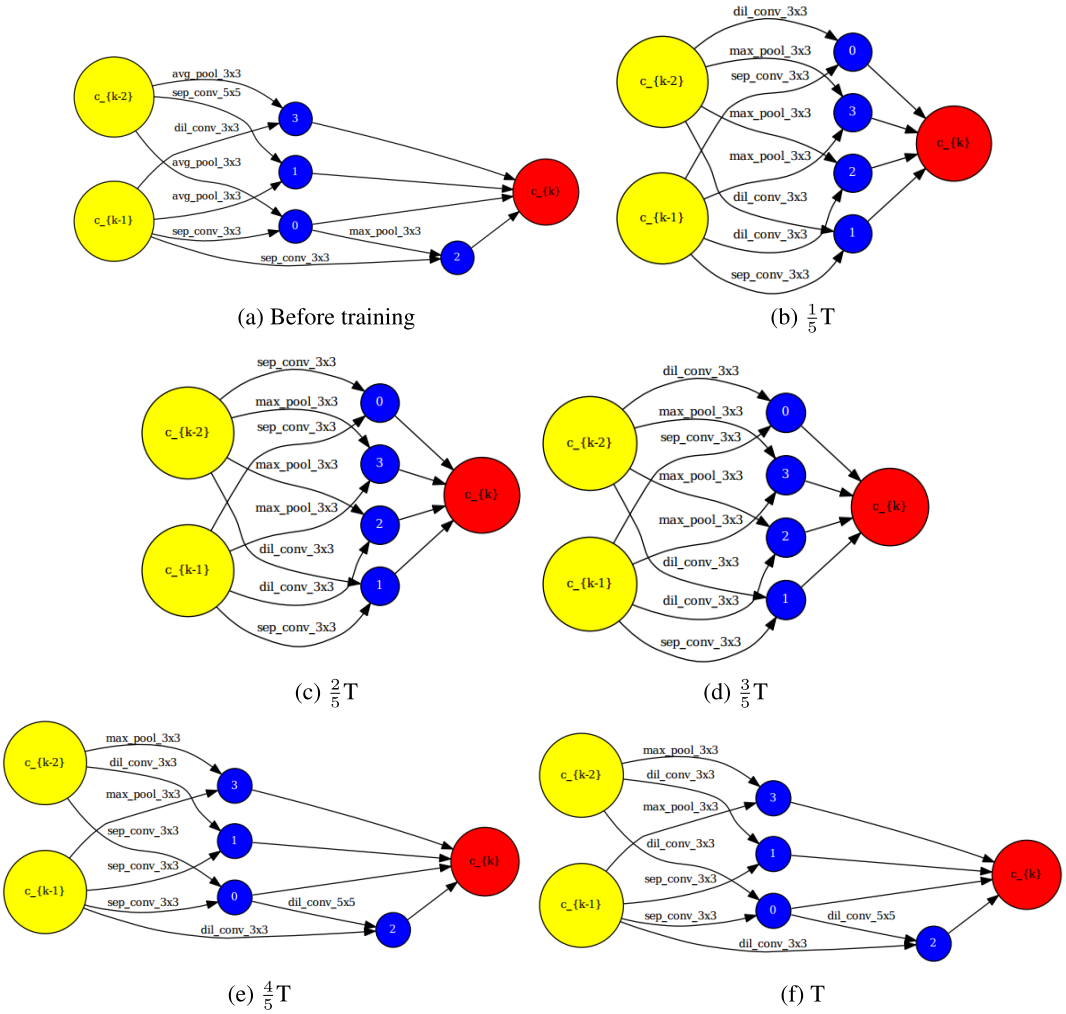


FIGURE 3. Change of the normal cell structure during the search stage of DARTS-CD. (T denotes the entire training process.)

TABLE 2. Parameter size and test accuracy of the models shown in FIGURE 3. The test accuracy is obtained by retraining each network for 600 epochs as in the evaluation stage.

Architecture	(a)	(b)	(c)	(d)	(e)	(f)
Parameter size (MB)	2.89	2.89	3.07	2.89	3.27	3.09
Test acc. (%)	96.87	97.15	97.14	97.15	97.22	97.21

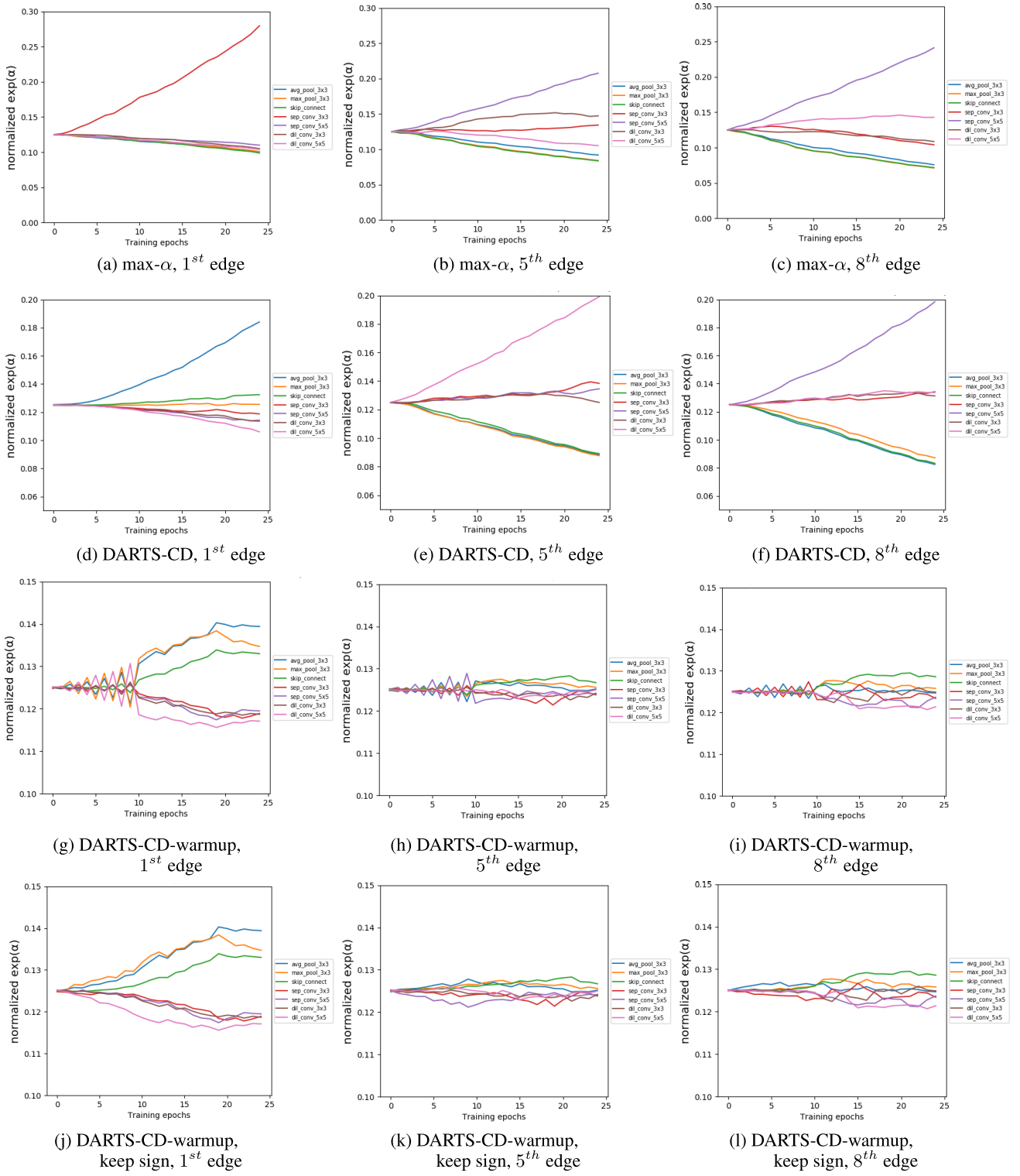
VI. DISCUSSION

A. COMPARISON WITH OTHER METHODS

In this subsection, we provide a comparison of DARTS-CD with several related studies in detail. We first compare DARTS-CD with DARTS in terms of the exploration of the search space. Then, DARTS-CD is compared with other methods in terms of path activation scheme, which represents how information flows through the network during the search, according to which the parameters are updated.

The primary difference of DARTS-CD from DARTS is that it performs optimization at only one edge in each training step, following the coordinate descent algorithm. Because DARTS-CD reduces the cell structure search problem to the operation decision problem within an edge, one could point out that the algorithm performs the search only within a small local region, compared with DARTS. However, we argue that this is not a critical problem in a differentiable architecture search. As experimentally analyzed by Li and Talwalkar [16], a random sampling-based search achieves comparable results to DARTS under the same time constraint. This indicates that although DARTS is based on continuous relaxation, there are numerous random initial points that show comparable performance to its solution. This observation supports that DARTS-CD can identify local optima comparable to DARTS in terms of network performance. This insight is also supported by several recent studies on local search [29], [30].

TABLE 3 and FIGURE 5 present the main difference of DARTS-CD from several NAS studies. The novelty of the proposed method is that edges use different path activation schemes that choose whether to perform only one operation per edge (single-path) or all candidate operations (multi-path) in a single forward pass. In TABLE 3,



(a) max-α, 1^{st} edge (b) max-α, 5^{th} edge (c) max-α, 8^{th} edge

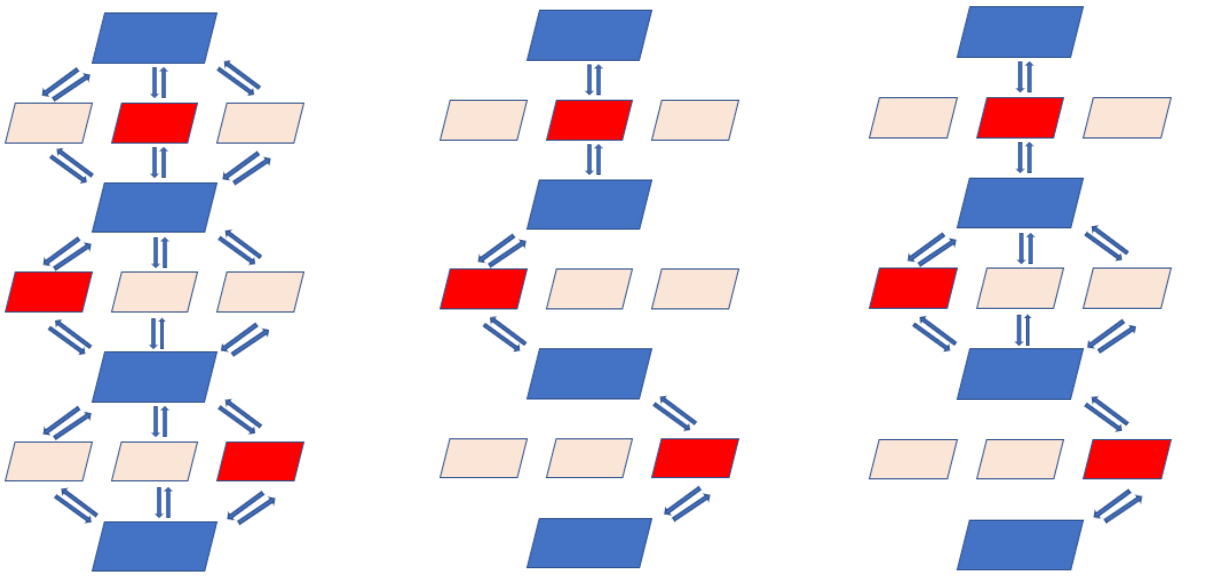
(d) DARTS-CD, 1^{st} edge (e) DARTS-CD, 5^{th} edge (f) DARTS-CD, 8^{th} edge

(g) DARTS-CD-warmup, 1^{st} edge (h) DARTS-CD-warmup, 5^{th} edge (i) DARTS-CD-warmup, 8^{th} edge

(j) DARTS-CD-warmup, keep sign, 1^{st} edge (k) DARTS-CD-warmup, keep sign, 5^{th} edge (l) DARTS-CD-warmup, keep sign, 8^{th} edge

FIGURE 4. Architecture parameter dynamics during the search stage of the methods introduced in Section III-B: max-α (a,b,c), DARTS-CD (d,e,f), and DARTS-CD-warmup (g,h,i). In (j,k,l), we present the same results as in (g,h,i) again with the architecture parameters inverted back to their original sign during warmup, to explicitly show the change in max-α operation.

according to previous methods, all the edges share the path activation scheme and possibly use different schemes to update the operation parameters and architecture parameters. For example, differentiable methods such as DARTS [14], SNAS [21], and P-DARTS [24] use multi-path activation. Other methods based on sampling, including NASNet [12]



(a) Multi-path forward and backward (b) Single-path forward and backward (c) Mixed-path forward and backward

FIGURE 5. Path activation types in TABLE 3 are visualized. Blue planes denote nodes, apricot planes denote operations, and red planes indicate operations selected by some policy (i.e., random sampling or max-α).

TABLE 3. Comparison of path activation scheme used for updating each parameter group in NAS methods.

Method	operation parameter update	architecture parameter update
DARTS [14]	multi-path	multi-path
SNAS [21]	multi-path	multi-path
P-DARTS [24]	multi-path	multi-path
NASNet [12]	single-path	single-path
AmoebaNet [13]	single-path	single-path
ENAS [23]	single-path	single-path
ProxylessNAS [25]	single-path	multi-path
DARTS-CD (ours)	multi-path for one edge, single-path for the others	

and AmoebaNet [13], use single-path activation. ProxylessNAS [25] uses single-path activation when updating the operation parameters and multi-path activation for architecture parameters. Still, it uses the same path activation scheme for all the edges in each training step. Only the operation parameters or architecture parameters are updated. Compared with other methods, DARTS-CD is the first to use a mixed path activation scheme, which leverages the efficiency of the differentiable search while maintaining the interpretability of the derivation process.

There are many possible intermediate path activation schemes between the proposed method and DARTS. For example, a multi-path activation scheme can be applied to M (M > 1) sampled edges, where M can be fixed or varied during training. This type of variant can have better exploration ability than the current version of DARTS-CD because search parameters are optimized for more edges in a single epoch. However, we choose to use multi-path activation in only one edge to spare search cost in terms of memory and computation, and propose a regularization method to improve the exploration ability of the proposed method.

In addition, the proposed method samples a single edge from uniform distribution based on the assumption that every edge is of equal importance. However, more effective sampling strategies can be used. For example, if the contribution of each edge to the network performance can be measured, it can be used as a sampling probability to give more search opportunities to more important edges. Therefore, our future work involves determining improved edge sampling strategies.

B. MODEL INCONSISTENCY FOR SEARCH AND EVALUATION

The one-shot network trained in the search stage of DARTS has a large gap from the derived network. In Xie *et al.* [21], this gap is attributed to the derivation process of DARTS, which removes most of the operations and edges, thereby changing the output of the network significantly. If this gap is critical in the forward pass of the network, it would also affect the gradient. Assuming that the gradients of specific operation parameters are affected by others, the speed and the performance of the optimization process would be degraded.

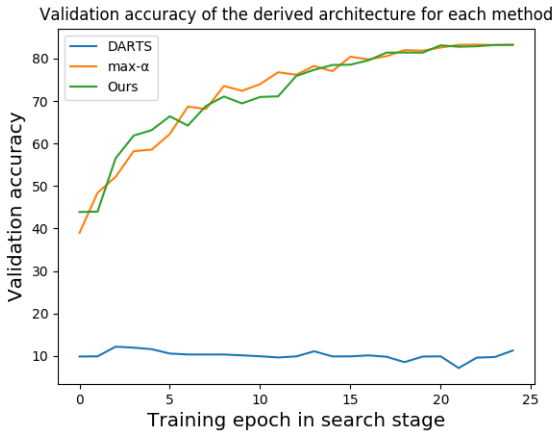


FIGURE 6. **Validation accuracy of the derived model at every training step during the *search stage* for DARTS, max $-\alpha$, and DARTS-CD (ours). The same derivation algorithm, based on architecture parameters, was applied to each method.**

We focus on the perspective that gradient descent (stochastic, batch, etc.) is not the only approach to train a neural network. Recent deep learning algorithms mostly use gradient descent as the optimizer. Following early deep learning studies that mostly used stochastic (batch) gradient descent, researchers have suggested improved optimization methods such as RMSprop, Adagrad [31], and Adam [32]. Although these methods have successfully trained large-scale deep neural networks, some other methods are known to be better for solving specific problems. One example is the coordinate descent algorithm, in which one parameter is updated per training step. This is a considerable difference from gradient descent-based algorithms, which update all the parameters collectively. Because coordinate descent searches for the optimal point along one axis in each training step, it serves as an effective optimization method when the parameters are independent of others. When the algorithm is applied to regression problems such as LASSO, it not only leads to fast convergence but also provides state-of-the-art optimization results [17], [20].

We suggest leveraging the coordinate descent algorithm for one-shot architecture search. In the one-shot network, the number of operation parameters is quite larger than that of ordinary neural networks. Thus, if all the operation parameters contribute to the gradient calculation as in gradient descent algorithms, it could lead to heavy dependence among the operations. This is verified by the near-random performance of the derived model with the operation parameters trained in the search stage (see FIGURE 6, blue line). Therefore, the one-shot networks must not be optimized in the same manner as training ordinary CNNs. According to the coordinate descent algorithm, we propose a novel training strategy in which only the parameters at one edge are optimized at each training step. In this framework, the parameters are trained considering that the other edges perform exactly the same as the derived network, so that the discrepancy of network behavior is minimal between the search and evaluation stages. Thus, all the candidate operations adjust better to the derivation of operations in other edges, which leads to expected performance curve, as shown in FIGURE 6.

VII. CONCLUSION

In this study, we propose DARTS-CD, an efficient differentiable architecture search algorithm based on the coordinate descent. In DARTS-CD, only one edge inside the cell is sampled at every training step for differentiable training, whereas one operation is chosen for the other edges. This modification to DARTS improves performance with only 1/12 relative search cost. Through experimental analysis, we show that DARTS-CD achieves satisfactory performance and trains the operation parameters so that they can perform well even in the derived model. Additionally, we suggest a warmup regularization to alleviate the low explorative power of sampling-based methods. This technique enhances the performance while showing the desired search behavior of differentiable search algorithms.

This study proposes an improvement to DARTS, which can be further extended in several aspects. First, we plan to apply DARTS-CD to the second-order version of DARTS, which is expected to benefit more from the efficiency of coordinate descent algorithm. The multi-path activation scheme can be applied to multiple edges instead of only one edge with only a small increase of computational cost. This extension is expected to improve the exploration ability of the proposed method. Furthermore, DARTS-CD can be widely applied to other differentiable search methods, by combining coordinate descent and NAS algorithms.

ACKNOWLEDGMENT

(Pyunghwan Ahn and Hyeong-Gwon Hong are co-first authors.)

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[12] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.
[13] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4780–4789.
[14] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–13.
[15] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 549–558.
[16] L. Li and A. Talwalkar, "Random search and reproducibility for neural architecture search," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 367–377.
[17] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM J. Optim.*, vol. 22, no. 2, pp. 341–362, Jan. 2012.
[18] Q. Deng and C. Lan, "Efficiency of coordinate descent methods for structured nonconvex optimization," 2019, *arXiv:1909.00918*. [Online]. Available: http://arxiv.org/abs/1909.00918
[19] A. Patrascu and I. Necoara, "Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization," *J. Global Optim.*, vol. 61, no. 1, pp. 19–46, Jan. 2015.
[20] Y. Fujiwara, Y. Ida, H. Shiokawa, and S. Iwamura, "Fast lasso algorithm via selective coordinate descent," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, p. 1561–1567.
[21] S. Xie, H. Zheng, C. Liu, and L. Lin, "SNAS: Stochastic neural architecture search," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.
[22] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: https://arxiv.org/abs/1605.07146
[23] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4095–4104.
[24] X. Chen, L. Xie, J. Wu, and Q. Tian, "Progressive differentiable architecture search: Bridging the depth gap between search and evaluation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1294–1303.
[25] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
[26] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*. [Online]. Available: http://arxiv.org/abs/1708.04552
[27] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
[28] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.
[29] C. White, S. Nolen, and Y. Savani, "Local search is state of the art for neural architecture search benchmarks," 2020, *arXiv:2005.02960*. [Online]. Available: http://arxiv.org/abs/2005.02960
[30] T. Den Ottelander, A. Dushatskiy, M. Virgolin, and P. A. N. Bosman, "Local search is a remarkably strong baseline for neural architecture search," 2020, *arXiv:2004.08996*. [Online]. Available: http://arxiv.org/abs/2004.08996
[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

PYUNGHWAN AHN received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include computer vision and deep learning.

HYEONG GWON HONG received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2019 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Graduate School of AI. His research interests include computer vision and deep learning.

JUNMO KIM (Member, IEEE) received the B.S. degree from Seoul National University, Seoul, South Korea, in 1998, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 2000 and 2005, respectively. From 2005 to 2009, he was a Research Staff Member with Samsung Advanced Institute of Technology (SAIT), South Korea. He joined the Faculty of Korea Advanced Institute of Science and Technology (KAIST), in 2009, where he is currently an Associate Professor of electrical engineering. His research interests include image processing, computer vision, statistical signal processing, machine learning, and information theory.