Auto-DeepLab

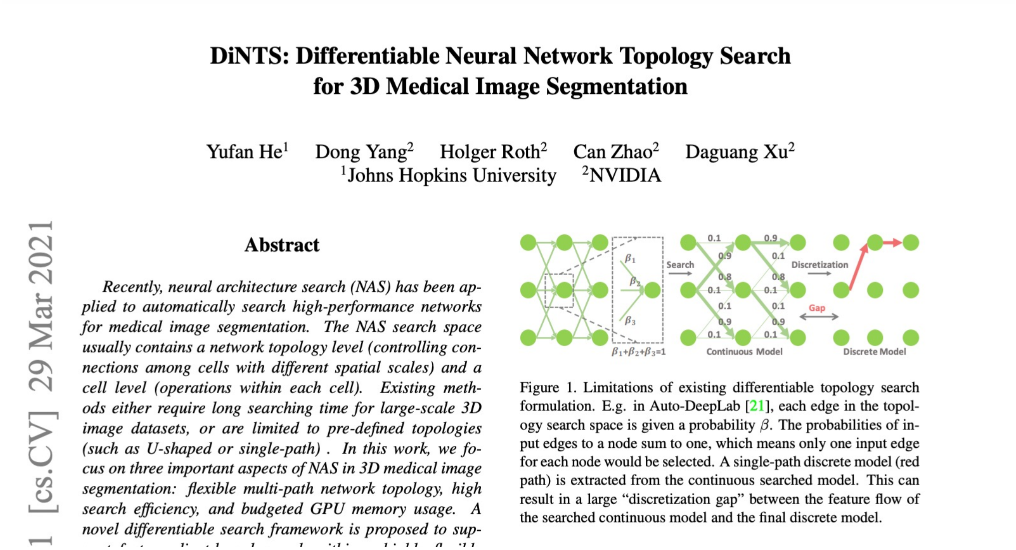
Hierarchical Neural Architecture Search for Semantic Image Segmentation

Image Processing Team
Seonok Kim(🙋‍♀️)

CVPR 2019 (Oral Presentation)
Authors: Chenxi Liu[1], Liang-Chieh Chen[2], Florian Schroff[2], Hartwig Adam[2], Wei Hua[2], Alan Yuille[1], Li Fei-Fei[3]
[1]Johns Hopkins University, [2]Google, [3]Stanford University

Auto-DeepLab for 3D Medical Image

(CVPR 2021, Oral) DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation



DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation

Yufan He[1] Dong Yang[2] Holger Roth[2] Can Zhao[2] Daguang Xu[2]
[1]Johns Hopkins University [2]NVIDIA

Abstract

Recently, neural architecture search (NAS) has been applied to automatically search high-performance networks for medical image segmentation. The NAS search space usually contains a network topology level (controlling connections among cells with different spatial scales) and a cell level (operations within each cell). Existing methods either require long searching time for large-scale 3D image datasets, or are limited to pre-defined topologies (such as U-shaped or single-path) . In this work, we focus on three important aspects of NAS in 3D medical image segmentation: flexible multi-path network topology, high search efficiency, and budgeted GPU memory usage. A novel differentiable search framework is proposed to sup-

Figure 1. Limitations of existing differentiable topology search formulation. E.g. in Auto-DeepLab [21], each edge in the topology search space is given a probability β. The probabilities of input edges to a node sum to one, which means only one input edge for each node would be selected. A single-path discrete model (red path) is extracted from the continuous searched model. This can result in a large "discretization gap" between the feature flow of the searched continuous model and the final discrete model.

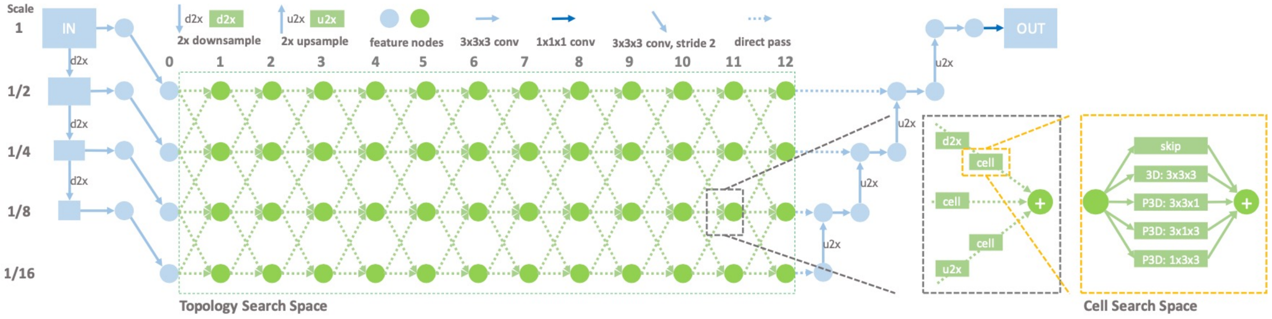


Figure 2. Our search space contains L=12 layers. The blue edges are the stem containing pre-defined operations. The cell operations are defined on the edges while the nodes are feature maps. Edges in the topology search space that are selected for features to flow from input to output form a candidate network topology. Each edge in the search space includes a cell which contains O=5 operations to select from. A downsample/upsample edge also contains a 2× downsample/upsample operation.

- A Differentiable Network Topology Search scheme DiNTS, which supports more flexible topologies and joint two-level search.
- A topology guaranteed discretization algorithm and a discretization aware topology loss for the search stage to minimize the discretization gap
- A memory usage aware search method which is able to search 3D networks with different GPU memory requirements.
  - 5.8 GPU days (recent C2FNAS takes 333 GPU days on the same dataset)

Introduction

Auto-DeepLab

Neural Architecture Search(NAS) for semantic image segmentation.

AutoML

Architecture Search

+

Segmentation

Dense image prediction

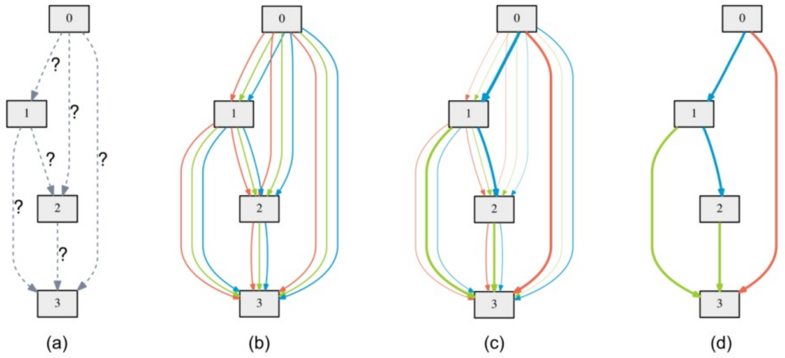
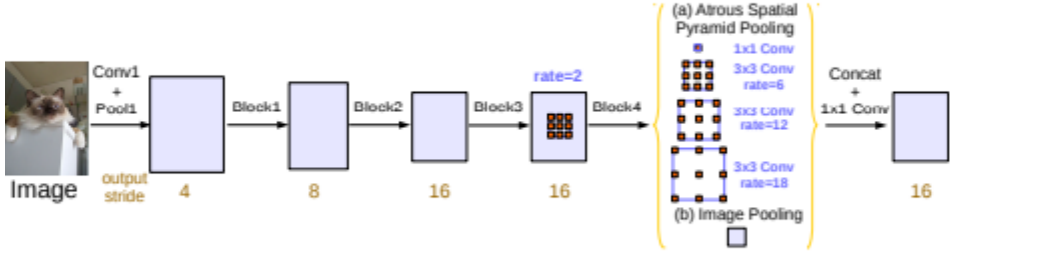


Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

DARTS (2018)[1]



DeepLab V3(2017)[2]

[1] Darts: Differentiable architecture search. arXiv:1806.09055, 2018

[2] Rethinking Atrous Convolution for Semantic Image Segmentation arXiv:1706.05587v3, 2017

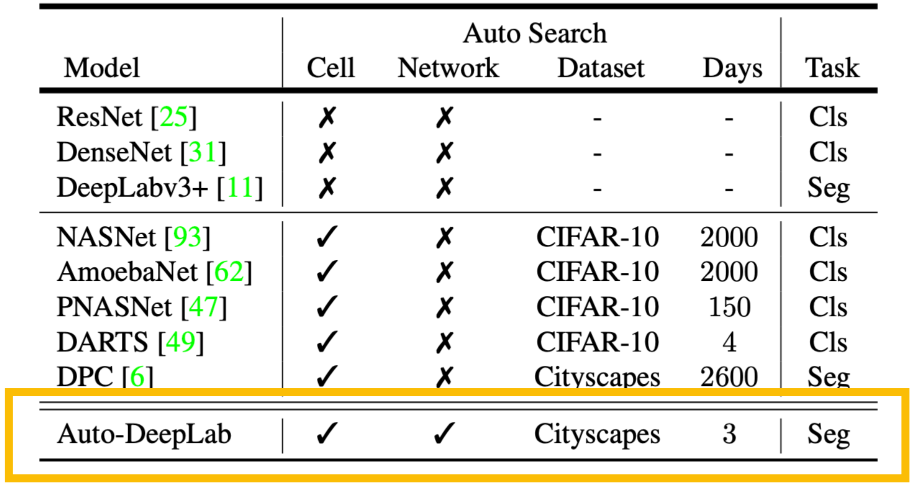
Contribution

One of the first attempts to extend NAS beyond image classification to dense image prediction.

The network level architecture search space that augments and complements the much-studied cell level one.

Efficient gradient-based architecture search (3 P100 GPU days on Cityscapes images).

The architecture for semantic image segmentation attains state-of-the-art performance without any ImageNet pretraining.



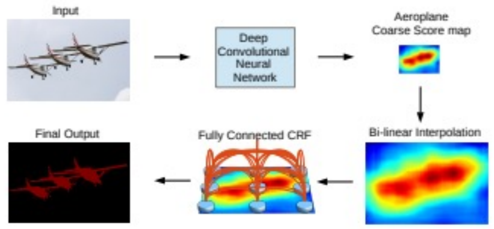
	Auto Search				
Model	Cell	Network	Dataset	Days	Task
ResNet [25]	✗	✗	-	-	Cls
DenseNet [31]	✗	✗	-	-	Cls
DeepLabv3+ [11]	✗	✗	-	-	Seg
NASNet [93]	✓	✗	CIFAR-10	2000	Cls
AmoebaNet [62]	✓	✗	CIFAR-10	2000	Cls
PNASNet [47]	✓	✗	CIFAR-10	150	Cls
DARTS [49]	✓	✗	CIFAR-10	4	Cls
DPC [6]	✓	✗	Cityscapes	2600	Seg
Auto-DeepLab	✓	✓	Cityscapes	3	Seg

Table 1: Comparing our work against other CNN architectures with two-level hierarchy. The main differences include: (1) we directly search CNN architecture for semantic segmentation, (2) we search the network level architecture as well as the cell level one, and (3) our efficient search only requires 3 P100 GPU days.

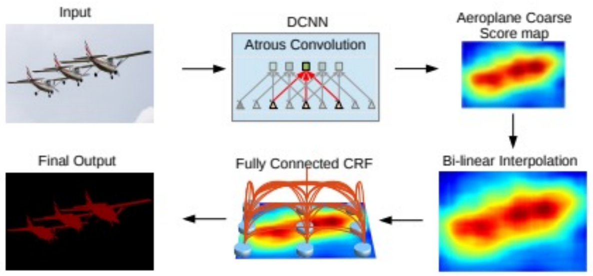
Related Work

Semantic Segmentation

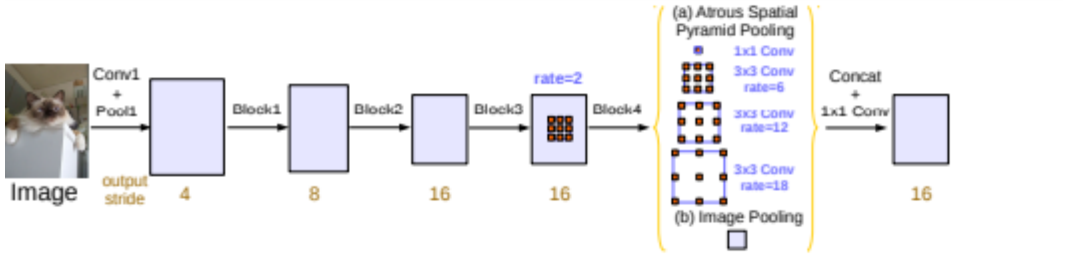
DeepLab V1 (2015)



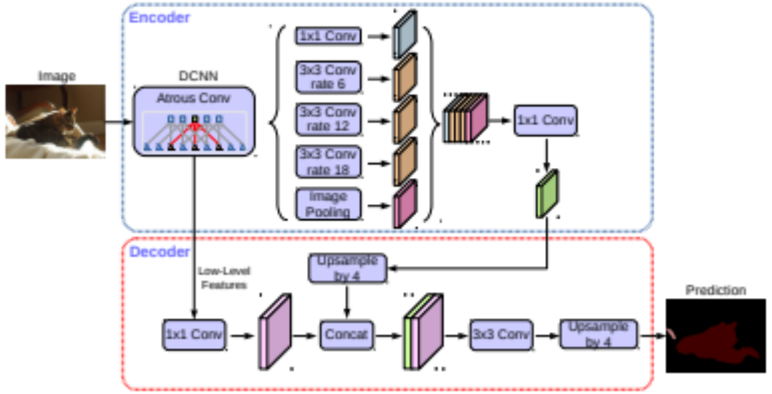
DeepLab V2 (2017)



DeepLab V3 (2017)



DeepLab V3+ (2018)



[1] Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. arXiv:1412.7062, 2015.
[2] DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. arXiv:1606.00915v2, 2017
[3] Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv:1706.05587v3, 2017
[4] Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. arXiv:1802.02611v3, 2018

ASPP(Atrous Spatial Pyramid Pooling)

ASSP is a semantic segmentation module for resampling a given feature layer at multiple rates prior to convolution

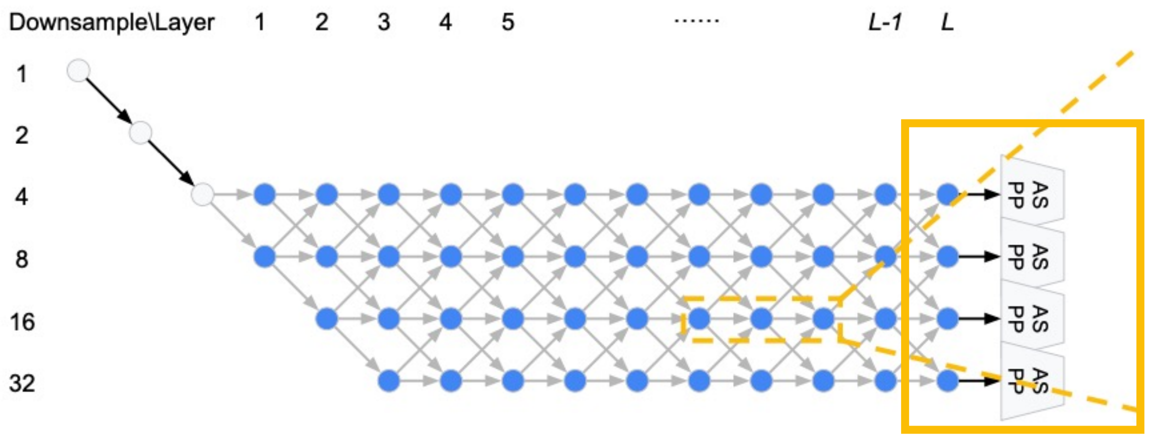


Figure 1: *Left:* Our network level search space with $L = 12$. Gray node
the blue nodes represents a candidate network level architecture. *Right:*
structure as described in Sec. 4.1.1. Every yellow arrow is associated
`concat` are associated with $\beta^l_{\frac{s}{2}\rightarrow s}, \beta^l_{s\rightarrow s}, \beta^l_{2s\rightarrow s}$ respectively, as descri

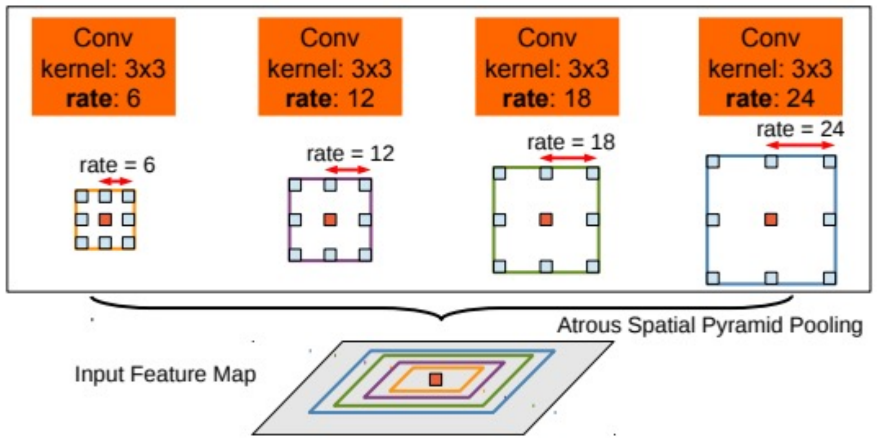


Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

ASPP[1]

[1] DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. arXiv:1606.00915v2, 2017

NAS (Network Architecture Search)

NAS aims at automatically designing neural network architectures, hence minimizing human hours and efforts. The authors' work follows the differentiable NAS formulation and extends it into the more general hierarchical setting.

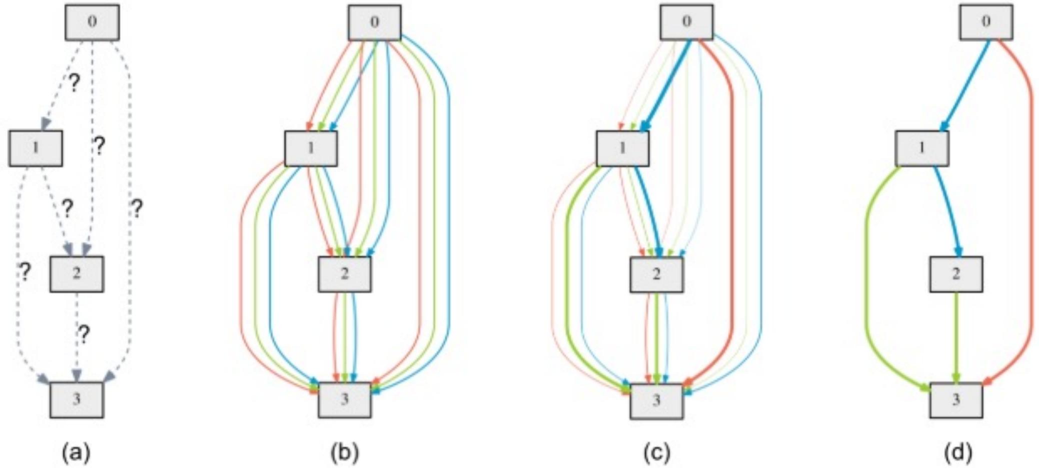


Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

DARTS (2018)[1]

[1] Darts: Differentiable architecture search. arXiv:1806.09055, 2018

Network level and Cell Level Search Space

The authors propose to search the network level structure in addition to the cell level structure, which forms a hierarchical architecture search space.

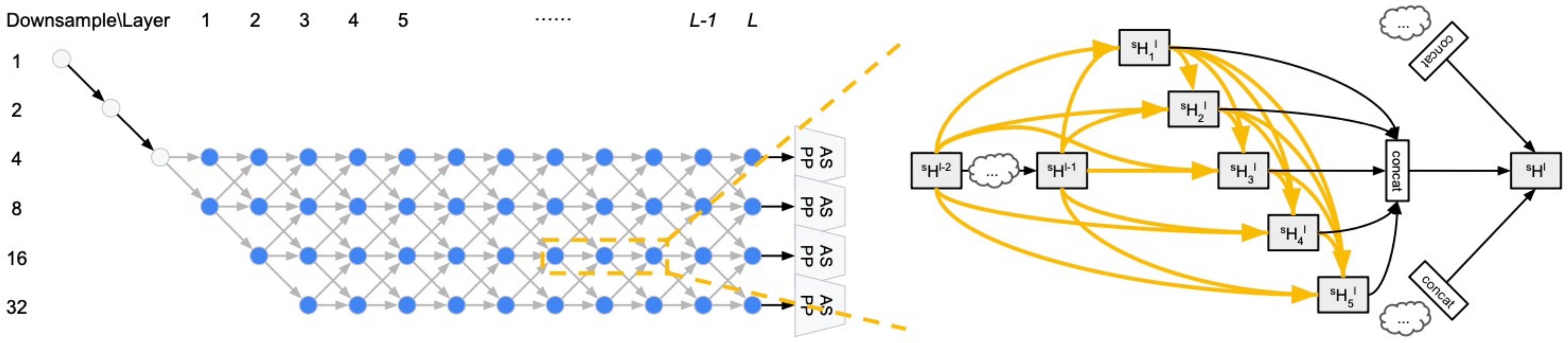
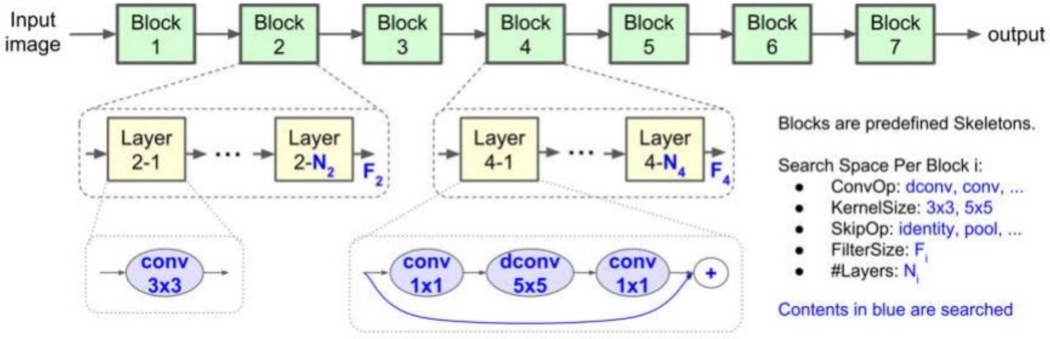


Figure 1: *Left:* Our network level search space with $L = 12$. Gray nodes represent the fixed "stem" layers, and a path along the blue nodes represents a candidate network level architecture. *Right:* During the search, each cell is a densely connected structure as described in Sec. 4.1.1. Every yellow arrow is associated with the set of values $\alpha_{j \to i}$. The three arrows after `concat` are associated with $\beta^l_{\frac{s}{2} \to s}, \beta^l_{s \to s}, \beta^l_{2s \to s}$ respectively, as described in Sec. 4.1.2. Best viewed in color.

Motivation



Blocks are predefined Skeletons.

Search Space Per Block i:

- ConvOp: dconv, conv, ...
- KernelSize: 3x3, 5x5
- SkipOp: identity, pool, ...
- FilterSize: F_i
- #Layers: N_i

Contents in blue are searched

MNasNet (2018)[1]

1024 * 2048 resolution

Cityspaces Dataset[2]

- Existing works often focus on searching the repeatable cell structure. The outer network level structure fixed by hand simplifies the search space.
- Optimal architectures for semantic segmentation must operate on high resolution imagery.
- Image classification should not be the end point for NAS.

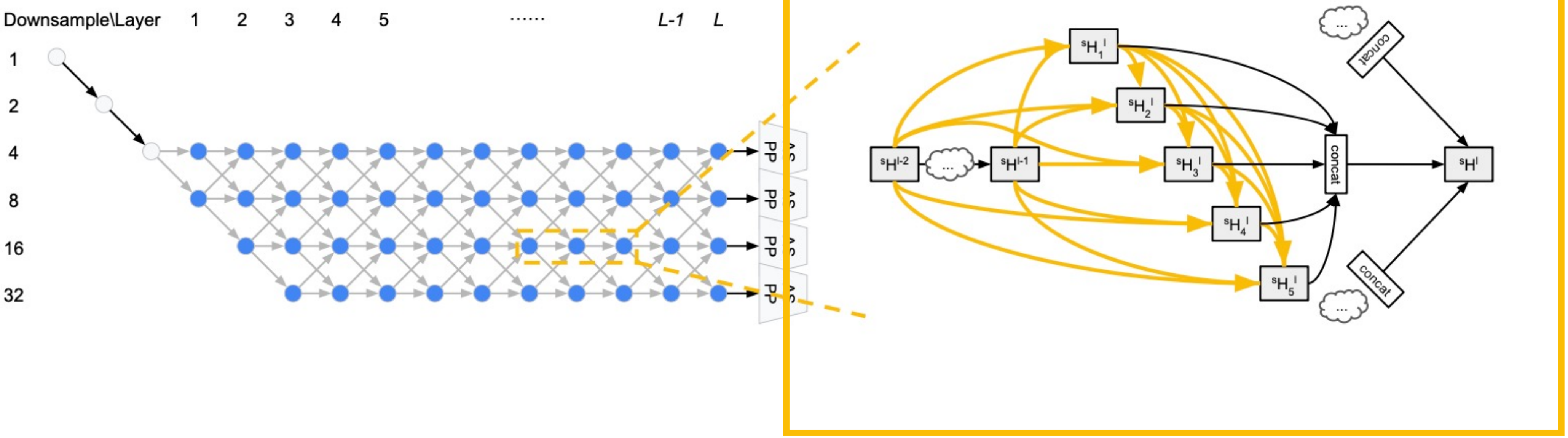
[1] MnasNet: Platform-Aware Neural Architecture Search for Mobile. arXiv:1807.11626, 2018

[2] www.cityscapes-dataset.com

Method

Cell Architecture

The authors reuse the continuous relaxation described in Darts[1].



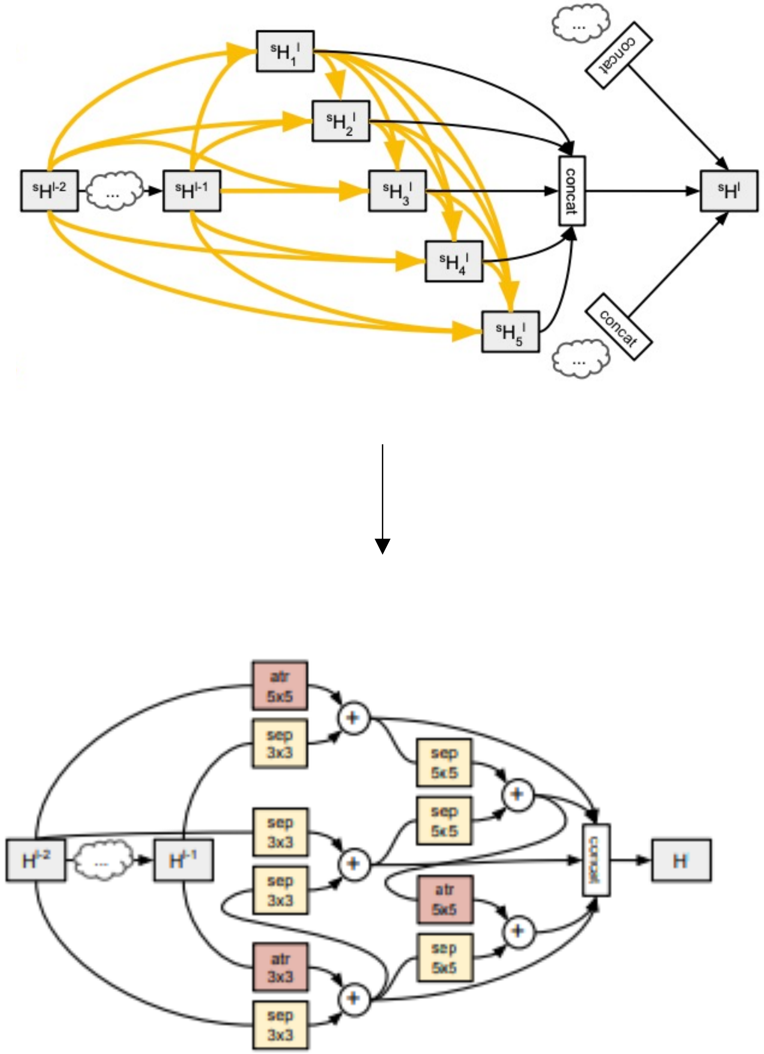
Cell Level Search Space

The set of possible layer types, O, consists of the following 8 operators, all prevalent in modern CNNs:

- 3 × 3 depthwise-separable conv
- 5 × 5 depthwise-separable conv
- 3×3 atrous conv with rate2
- 5×5 atrous conv with rate2
- 3 × 3 average pooling
- 3 × 3 max pooling
- skip connection
- no connection(zero)

Before
architecture search

After
architecture search



[1] Darts: Differentiable architecture search. arXiv:1806.09055, 2018

Cell Architecture

Every block’s output tensor Hil is connected to all hidden states in T_i^l.

$$H_i^l = \sum_{H_j^l \in T_i^l} O_{j\to i}(H_j^l)$$

$$\bar{O}_{j\to i}(H_j^l) = \sum_{O^k \in O} \alpha_{j\to i}^k O^k \, (H_j^l)$$

where

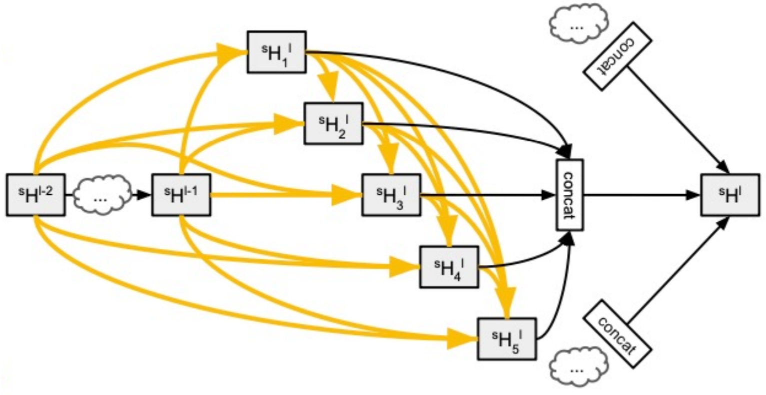
$$\sum_{k=1}^{|O|} \alpha_{j\to i}^k = 1 \qquad \alpha_{j\to i}^k \geq 0$$

H_i^l : output tensor at layer l

O : operator

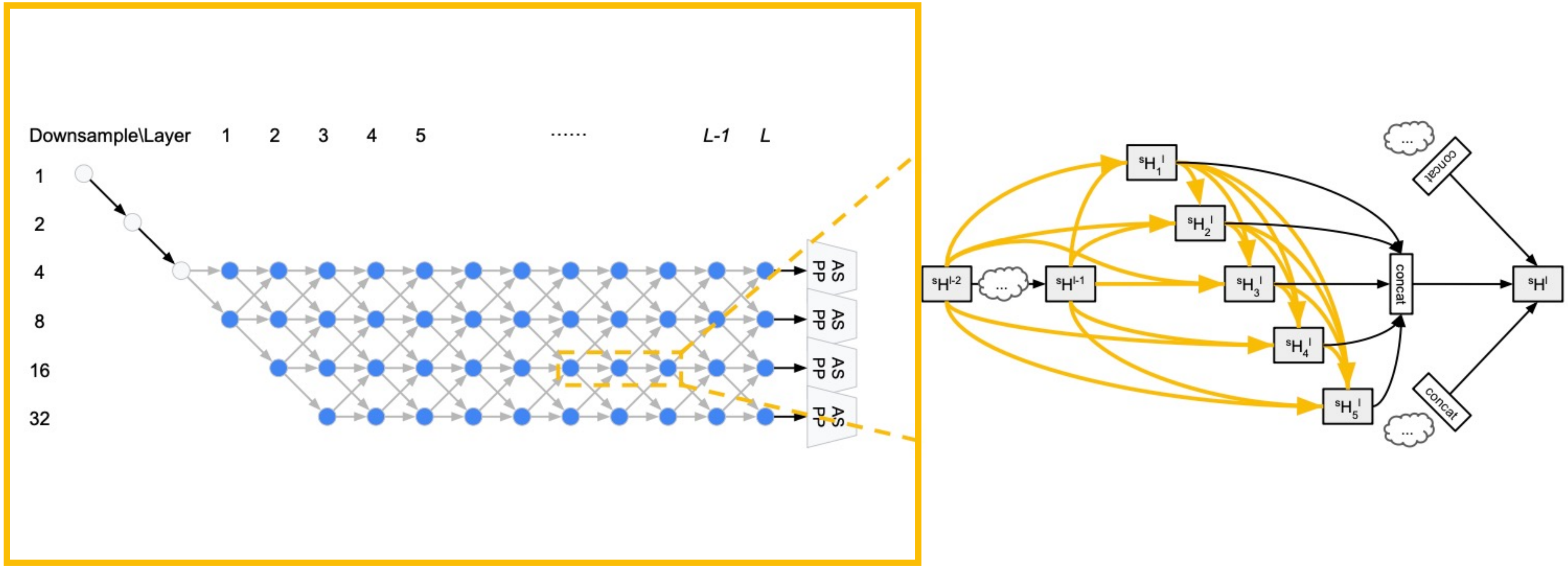
$\alpha_{j\to i}^k$: normalized scalars

$$H^l = \mathrm{Cell}(H^{l-1}, H^{l-2}; \alpha)$$

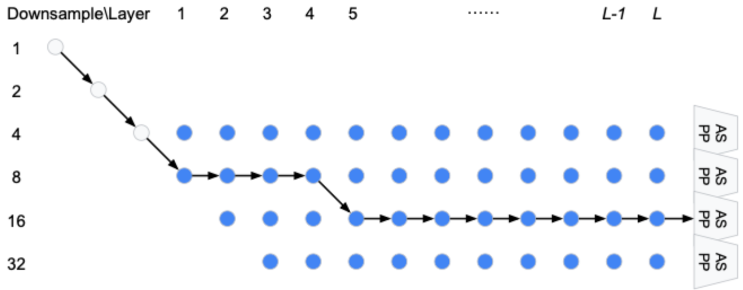


Network Architecture

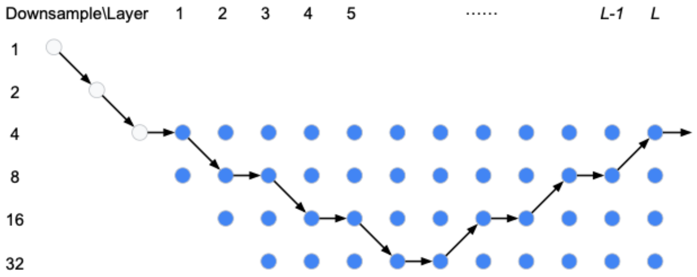
Within a cell, all tensors are of the same spatial size, which enables the (weighted) sum in equations of $H_i^l$and $\bar{O}_{j\to i}$.



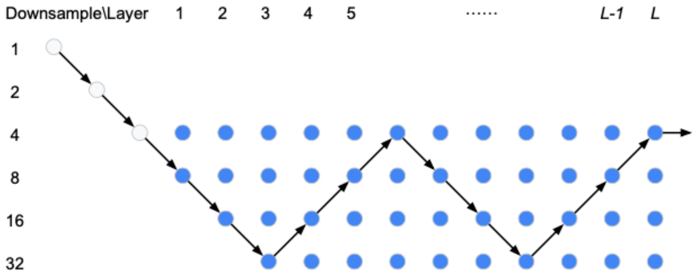
Network Level Search Space



(a) Network level architecture used in DeepLabv3 [9].



(b) Network level architecture used in Conv-Deconv [56].



(c) Network level architecture used in Stacked Hourglass [55].

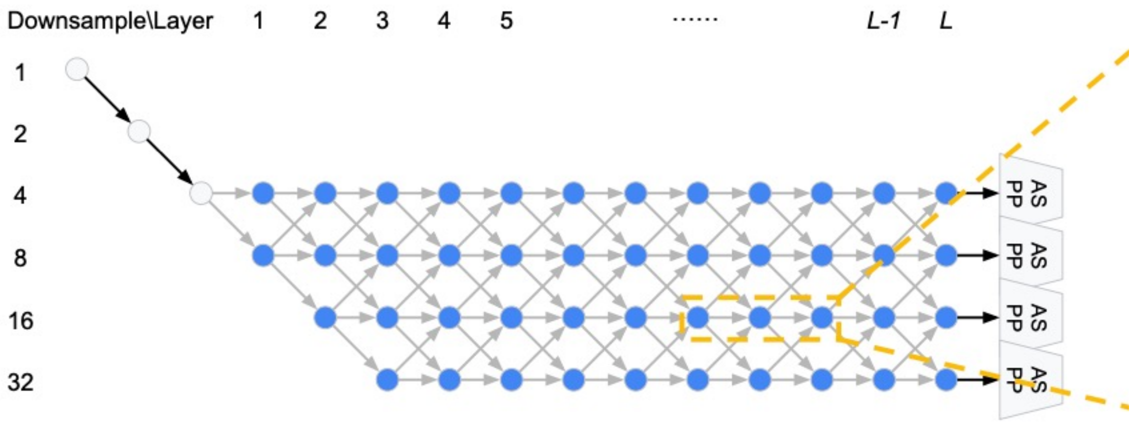
Two principles:

The spatial resolution of the next layer is either twice as large, or twice as small, or remains the same.

The smallest spatial resolution is down-sampled by 32.

Network Architecture

Each layer l will have at most 4 hidden states $\{{}^4H^l, {}^8H^l, {}^{16}H^l, {}^{32}H^l\}$, with the upper left superscript indicating the spatial resolution.



$$
\begin{aligned}
{}^sH^l = & \beta^l_{\frac{s}{2}\to s}\mathrm{Cell}({}^{\frac{s}{2}}H^{l-1}, {}^sH^{l-2}; \alpha) \\
& + \beta^l_{s\to s}\mathrm{Cell}({}^sH^{l-1}, {}^sH^{l-2}; \alpha) \\
& + \beta^l_{2s\to s}\mathrm{Cell}({}^{2s}H^{l-1}, {}^sH^{l-2}; \alpha)
\end{aligned}
$$

$$
\beta^l_{s\to\frac{s}{2}} + \beta^l_{s\to s} + \beta^l_{s\to 2s} = 1 \qquad \forall s, l
$$

$$
\beta^l_{s\to\frac{s}{2}} \geq 0 \quad \beta^l_{s\to s} \geq 0 \quad \beta^l_{s\to 2s} \geq 0 \qquad \forall s, l
$$

Optimization

The authors use 321 × 321 random image crops from half-resolution(512 × 1024) images in the train fine set and partition the training data into two disjoint sets train A and train B.

$trainA$ Update network weight w by $\nabla_w L_{trainA}(w, \alpha, \beta)$

$trainB$ Update architecture α, β by $\nabla_{\alpha,\beta} L_{trainB}(w, \alpha, \beta)$

Experiments

Architecture Search Implementation Details

A total of L = 12 layers in the network

321 × 321 random image crops from half-resolution (512 × 1024) images in the train fine set.

The architecture search optimization is conducted for a total of 40 epochs.

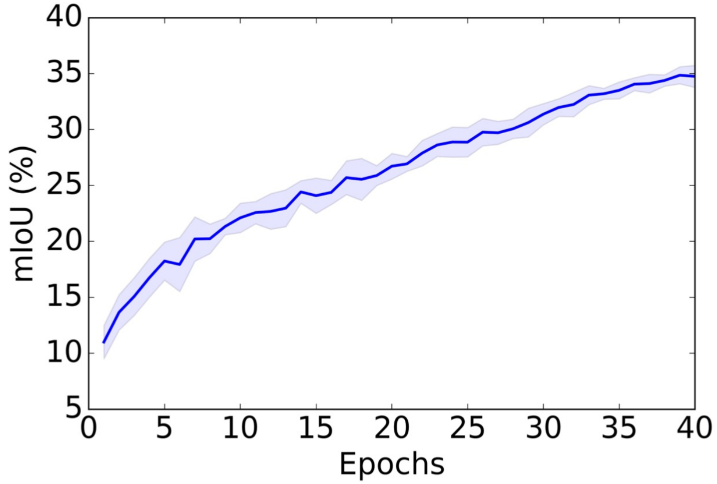


Figure 4: Validation accuracy during 40 epochs of architecture search optimization across 10 random trials.

Semantic Segmentation Results

The authors report their architecture search implementation details as well as the search results. The authors then report semantic segmentation results on benchmark datasets with our best-found architecture.

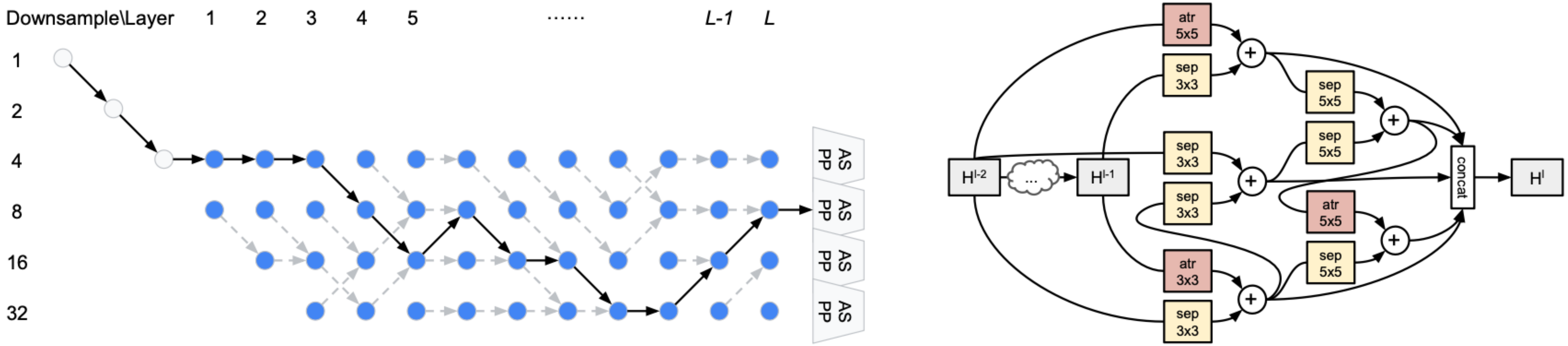


Figure 3: Best network and cell architecture found by our Hierarchical Neural Architecture Search. Gray dashed arrows show the connection with maximum β at each node. **atr:** atrous convolution. **sep:** depthwise-separable convolution.

Cityscapes

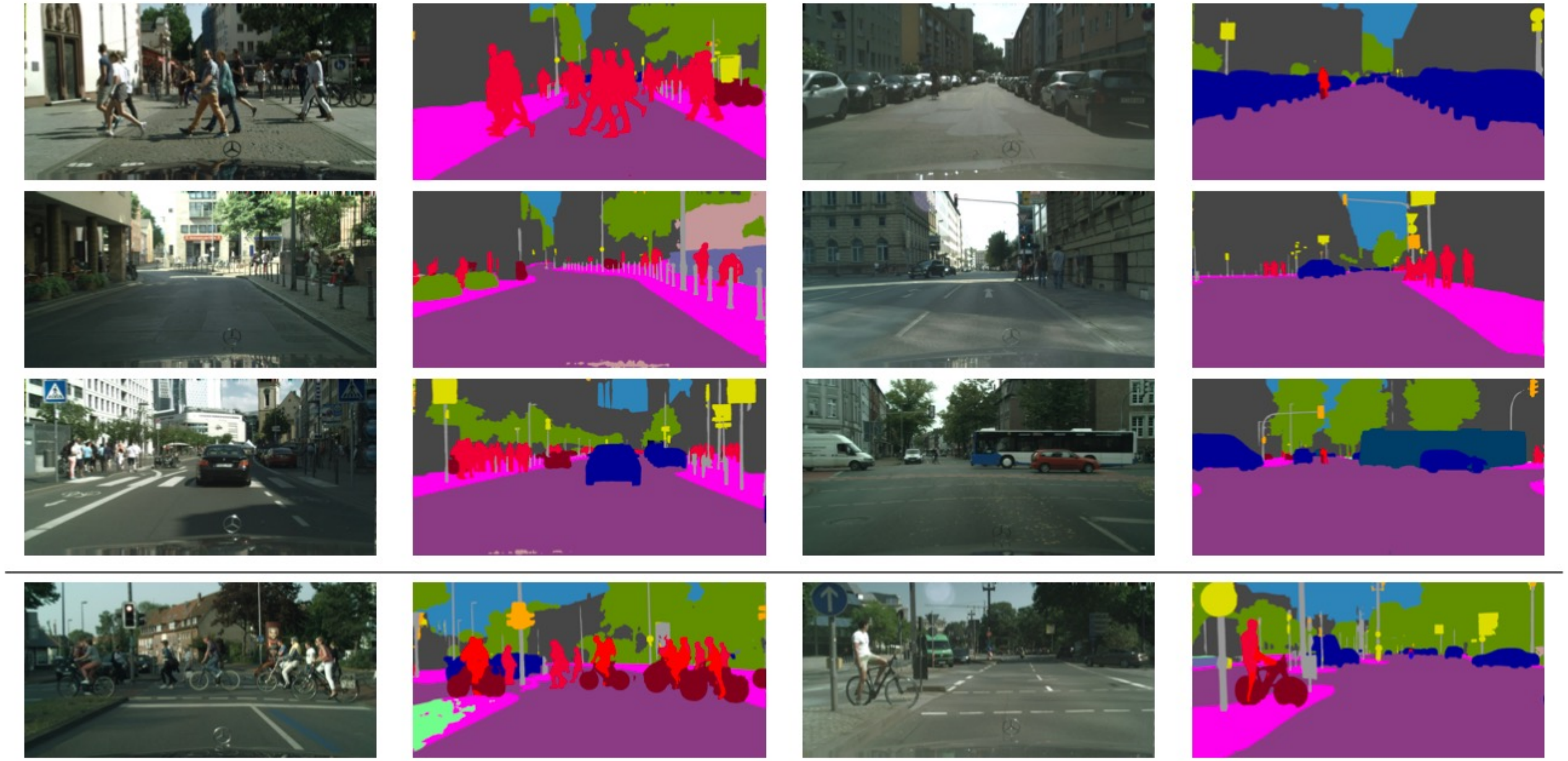
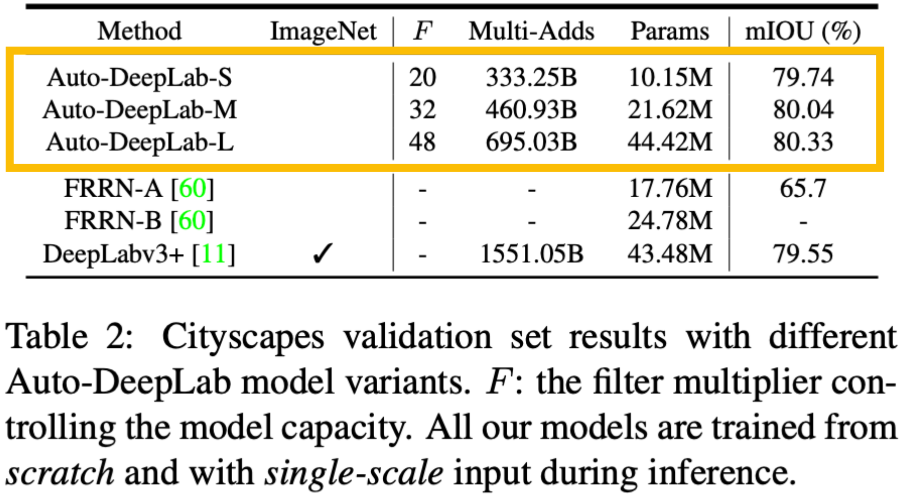


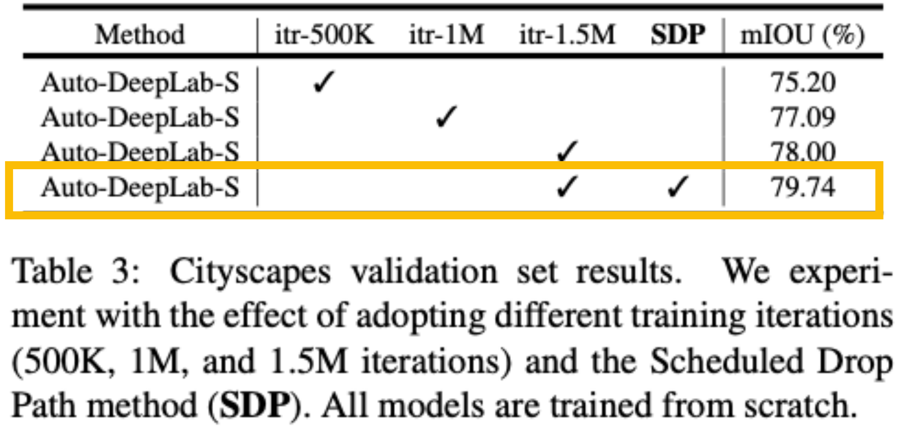
Figure 5: Visualization results on Cityscapes *val* set. Our failure mode is shown in the last row where our model confuses with some difficult semantic classes such as person and rider.

Cityscapes



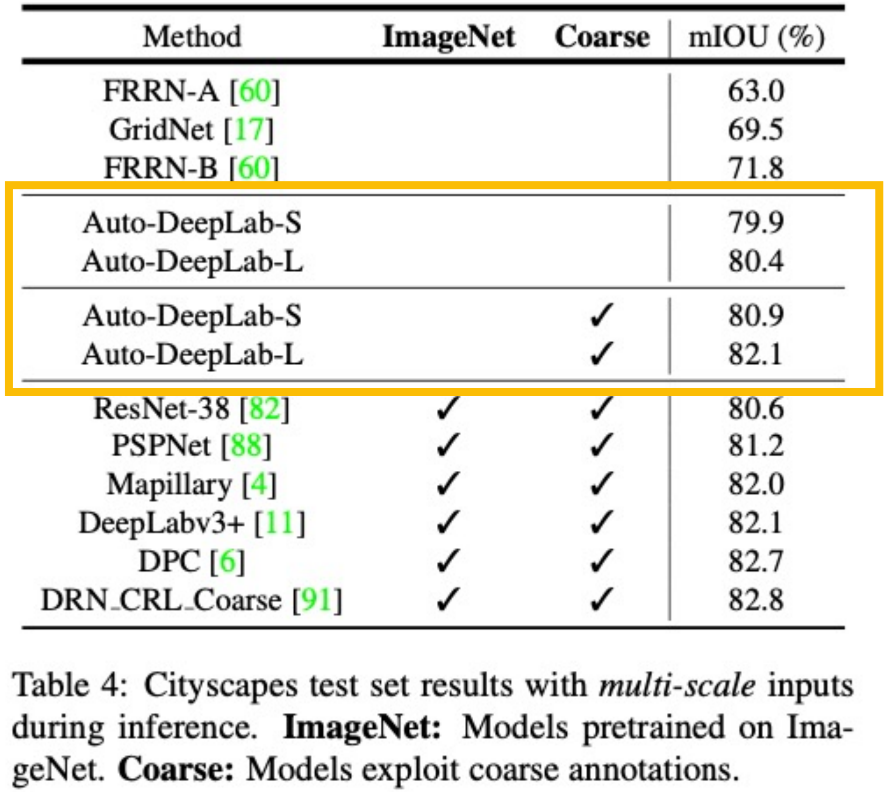
Method	ImageNet	F	Multi-Adds	Params	mIOU (%)
Auto-DeepLab-S		20	333.25B	10.15M	79.74
Auto-DeepLab-M		32	460.93B	21.62M	80.04
Auto-DeepLab-L		48	695.03B	44.42M	80.33
FRRN-A [60]		-	-	17.76M	65.7
FRRN-B [60]		-	-	24.78M	-
DeepLabv3+ [11]	✓	-	1551.05B	43.48M	79.55

Table 2: Cityscapes validation set results with different Auto-DeepLab model variants. F: the filter multiplier controlling the model capacity. All our models are trained from *scratch* and with *single-scale* input during inference.



Method	itr-500K	itr-1M	itr-1.5M	**SDP**	mIOU (%)
Auto-DeepLab-S	✓				75.20
Auto-DeepLab-S		✓			77.09
Auto-DeepLab-S			✓		78.00
Auto-DeepLab-S			✓	✓	79.74

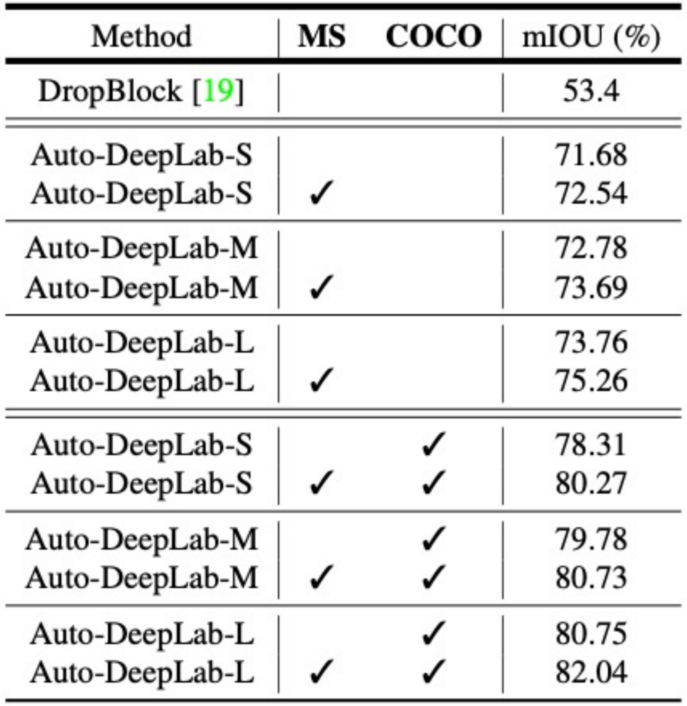
Table 3: Cityscapes validation set results. We experiment with the effect of adopting different training iterations (500K, 1M, and 1.5M iterations) and the Scheduled Drop Path method (**SDP**). All models are trained from scratch.



Method	**ImageNet**	**Coarse**	mIOU (%)
FRRN-A [60]			63.0
GridNet [17]			69.5
FRRN-B [60]			71.8
Auto-DeepLab-S			79.9
Auto-DeepLab-L			80.4
Auto-DeepLab-S		✓	80.9
Auto-DeepLab-L		✓	82.1
ResNet-38 [82]	✓	✓	80.6
PSPNet [88]	✓	✓	81.2
Mapillary [4]	✓	✓	82.0
DeepLabv3+ [11]	✓	✓	82.1
DPC [6]	✓	✓	82.7
DRN_CRL_Coarse [91]	✓	✓	82.8

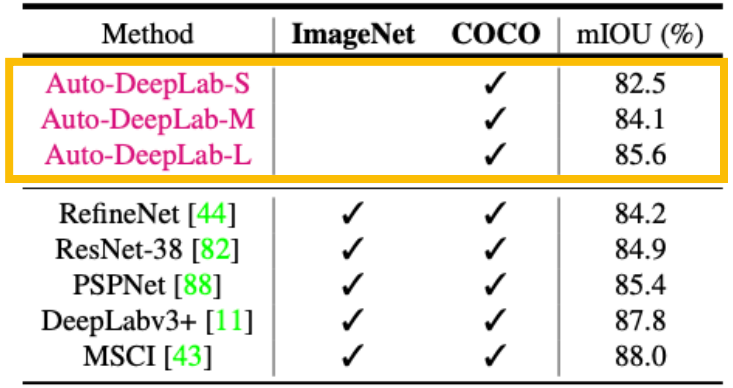
Table 4: Cityscapes test set results with *multi-scale* inputs during inference. **ImageNet:** Models pretrained on ImageNet. **Coarse:** Models exploit coarse annotations.

PASCAL VOC 2012



Method	**MS**	**COCO**	mIOU (%)
DropBlock [19]			53.4
Auto-DeepLab-S			71.68
Auto-DeepLab-S	✓		72.54
Auto-DeepLab-M			72.78
Auto-DeepLab-M	✓		73.69
Auto-DeepLab-L			73.76
Auto-DeepLab-L	✓		75.26
Auto-DeepLab-S		✓	78.31
Auto-DeepLab-S	✓	✓	80.27
Auto-DeepLab-M		✓	79.78
Auto-DeepLab-M	✓	✓	80.73
Auto-DeepLab-L		✓	80.75
Auto-DeepLab-L	✓	✓	82.04

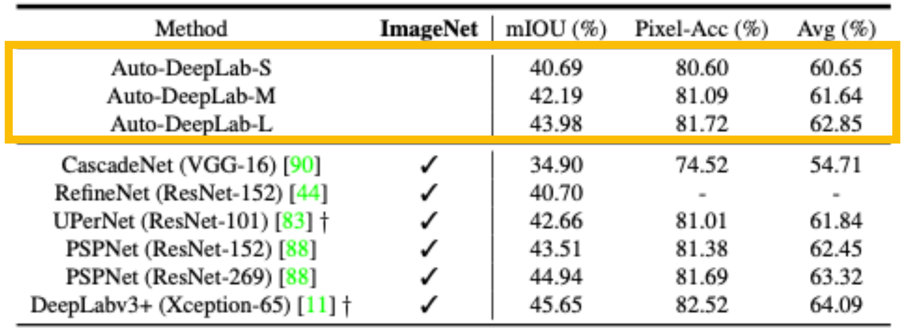
Table 5: PASCAL VOC 2012 validation set results. We experiment with the effect of adopting *multi-scale* inference (**MS**) and COCO-pretrained checkpoints (**COCO**). Without any pretraining, our best model (Auto-DeepLab-L) outperforms DropBlock by 20.36%. All our models are not pretrained with ImageNet images.



Method	**ImageNet**	**COCO**	mIOU (%)
Auto-DeepLab-S		✓	82.5
Auto-DeepLab-M		✓	84.1
Auto-DeepLab-L		✓	85.6
RefineNet [44]	✓	✓	84.2
ResNet-38 [82]	✓	✓	84.9
PSPNet [88]	✓	✓	85.4
DeepLabv3+ [11]	✓	✓	87.8
MSCI [43]	✓	✓	88.0

Table 6: PASCAL VOC 2012 test set results. Our Auto-DeepLab-L attains comparable performance with many state-of-the-art models which are pretrained on both **ImageNet** and **COCO** datasets. We refer readers to the official leader-board for other state-of-the-art models.

ADE20K



Method	**ImageNet**	mIOU (%)	Pixel-Acc (%)	Avg (%)
Auto-DeepLab-S		40.69	80.60	60.65
Auto-DeepLab-M		42.19	81.09	61.64
Auto-DeepLab-L		43.98	81.72	62.85
CascadeNet (VGG-16) [90]	✓	34.90	74.52	54.71
RefineNet (ResNet-152) [44]	✓	40.70	-	-
UPerNet (ResNet-101) [83] †	✓	42.66	81.01	61.84
PSPNet (ResNet-152) [88]	✓	43.51	81.38	62.45
PSPNet (ResNet-269) [88]	✓	44.94	81.69	63.32
DeepLabv3+ (Xception-65) [11] †	✓	45.65	82.52	64.09

Table 7: ADE20K validation set results. We employ *multi-scale* inputs during inference. †: Results are obtained from their up-to-date model zoo websites respectively. **ImageNet:** Models pretrained on ImageNet. Avg: Average of mIOU and Pixel-Accuracy.

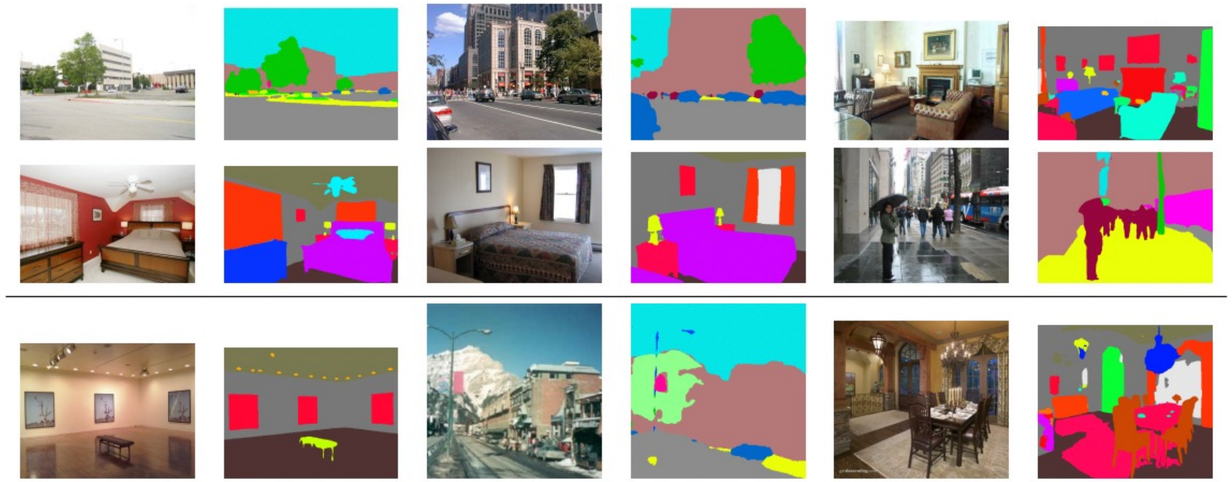
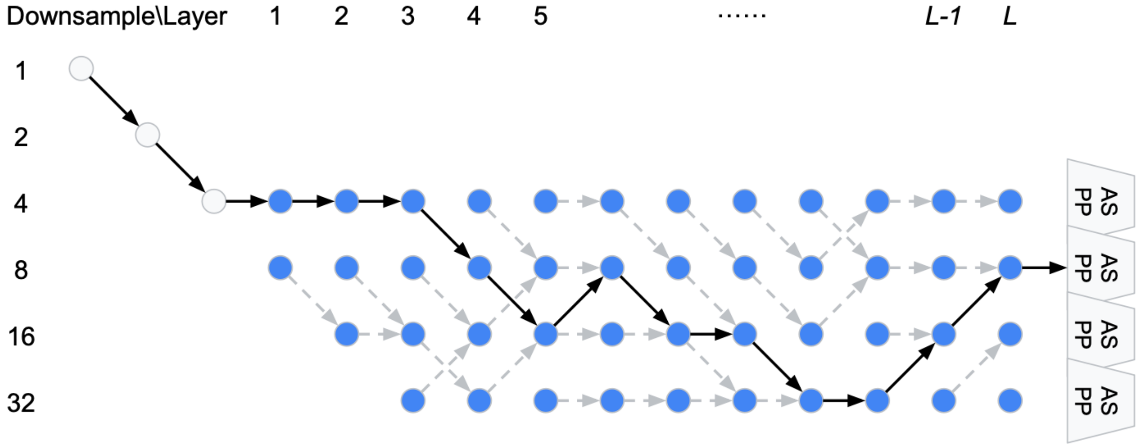


Figure 6: Visualization results on ADE20K *validation* set. Our failure mode is shown in the last row where our model could not segment very fine-grained objects (*e.g.*, chair legs) and confuse with some difficult semantic classes (*e.g.*, floor and rug).

Conclusion

Conclusion



The first attempts to extend Neural Architecture Search beyond image classification to dense image prediction problems.

A differentiable formulation that allows efficient (about 1000× faster than DPC) architecture search.

Auto-DeepLab significantly outperforms the previous state-of-the-art.

Sources

Paper
Auto-DeepLab
(https://arxiv.org/abs/1901.02985)

YouTube
PR-141: Auto-DeepLab
(https://youtu.be/ltlhQXHGzgE)

Blog
DeepLab V3+: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation (https://blog.lunit.io/2018/07/02/deeplab-v3-encoder-decoder-with-atrous-separable-convolution-for-semantic-image-segmentation/)

[Object Segmentation] ASPP : Atrous Spatial Pyramid Pooling
(https://eehoeskrap.tistory.com/459)

Differentiable Architecture Search (DARTS)
(https://ahn1340.github.io/neural%20architecture%20search/2021/05/03/DARTS)

Presenter

Seonok Kim (sokim0991@korea.ac.kr)