UNIVERSITÀ DEGLI STUDI FIRENZE
DIEF
Dipartimento di Ingegneria Industriale

Neural Networks for Semantic Segmentation

Machine Learning Course

Antonio Castellucci
Michela Crulli

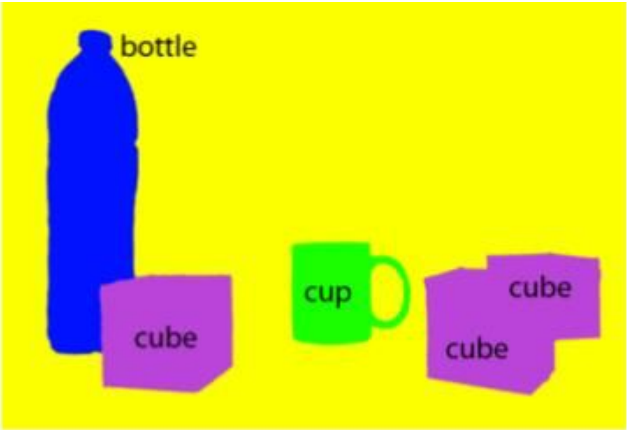
Overview

- Introduction
  - Semantic segmentation
  - Applications
  - Neural networks for semantic segmentation
- DUC + HDC
  - Approach
  - Results
- Deeplab V3+
  - Approach
  - Results
- ICNet
  - Approach
  - Results
- NAS
  - Introduction
  - DARTS
  - Auto-DeepLab
- Conclusions

- Semantic Segmentation
  - The goal of semantic image segmentation is to label *each pixel* of an image with a corresponding class of what is being represented where:

$$L = \{l_1, l_2, \dots, l_k\} \qquad X = \{x_1, x_2, \dots, x_N\}$$

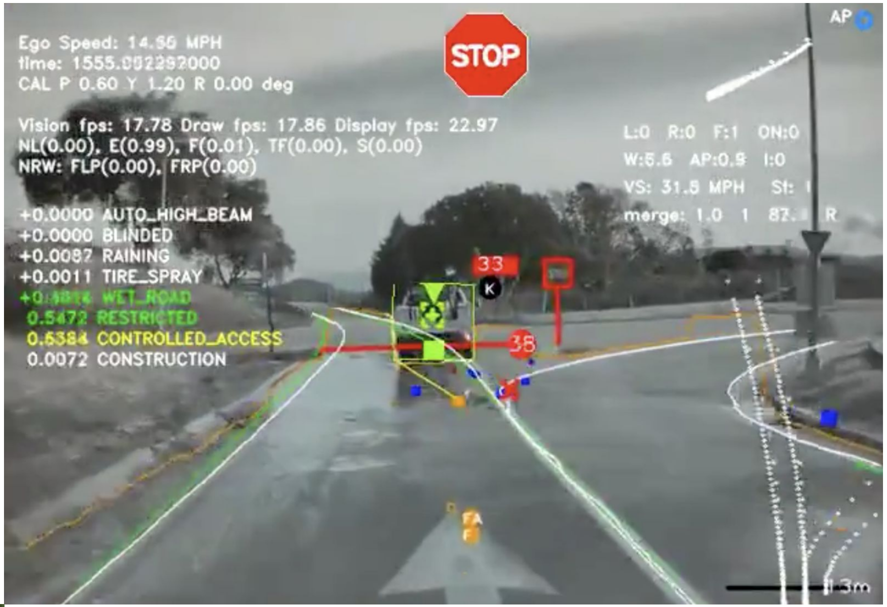
X is a ***2D*** image of ***W x H = N*** pixels ***x***



Introduction

- Applications

  - GeoSensing
  - Autonomous driving
  - Facial Segmentation
  - Precision Agriculture



Introduction

- Neural Networks for semantic segmentation

Before:

- Features hand-written
- Random Forest, Boosting
- Prediction of only central pixel of a patch

With gpu's growth:

- Neural Networks
- Convolutional Neural Networks

Introduction

- Neural Networks for semantic segmentation

3 key components:

- A fully convolutional network (FCN)
- Conditional Random Fields (CRF)
- Dilated convolution (Atrous convolution)

Common improvements:

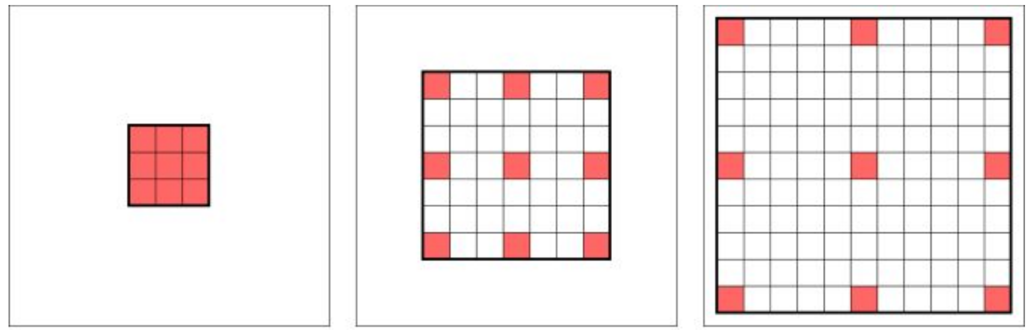
- Deeper FCN models
- More powerful CRFs

Understanding Convolution for Semantic Segmentation

(Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, Garrison Cottrel, 2018)

DUC + HDC: Encoding

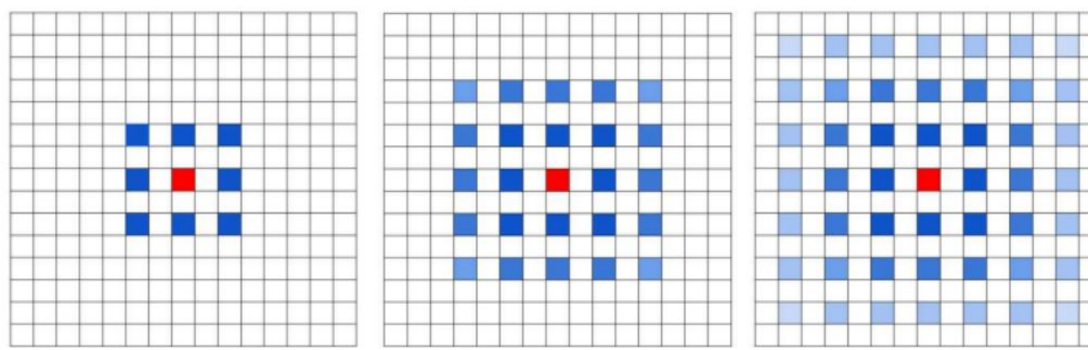
- Possible solution for denser feature maps:
  - Atrous convolution:



Kernel: 3x3 Dilation rate: 1 | Kernel: 3x3 Dilation rate: 3 | Kernel: 3x3 Dilation rate: 5

$$y[i] = \sum_{k} x[i + r \cdot k] w[k]$$

- Problem:
  - Gridding Problem

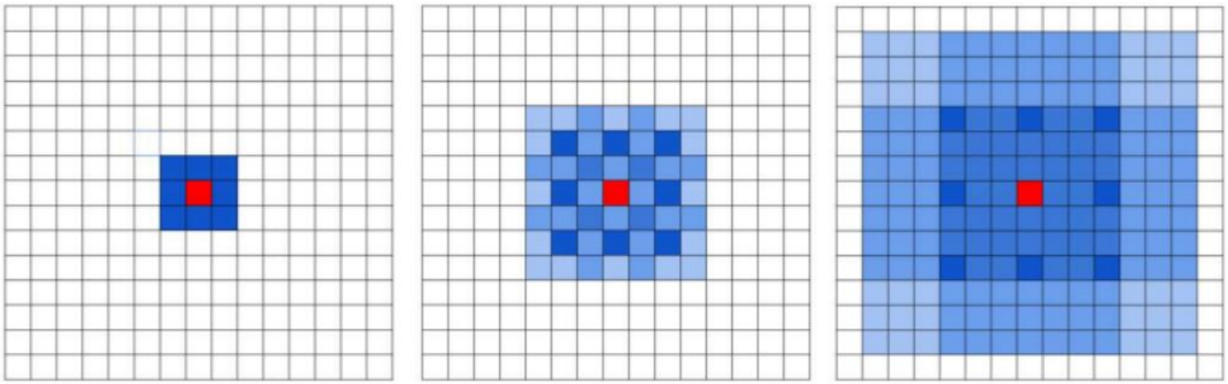


K = 3 | r = 2 | r = 2 | r = 2

DUC + HDC: Encoding

- Solution: Hybrid Dilated Convolution (HDC)

  - different dilatation rate r



r = 1 r = 2 r = 3

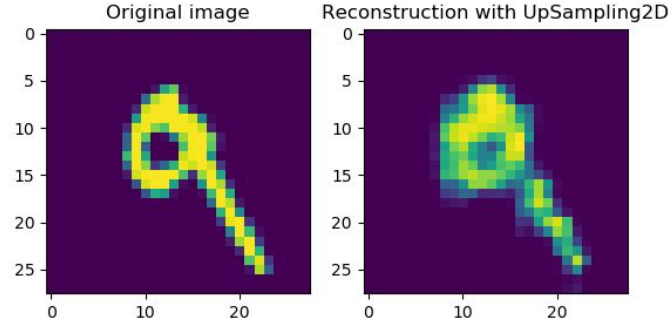
  - maximum distance between two nonzero values

$$M_i = \max[M_{i+1}-2r_i,\ M_{i+1}-2(M_{i+1}-r_i),\ r_i]$$

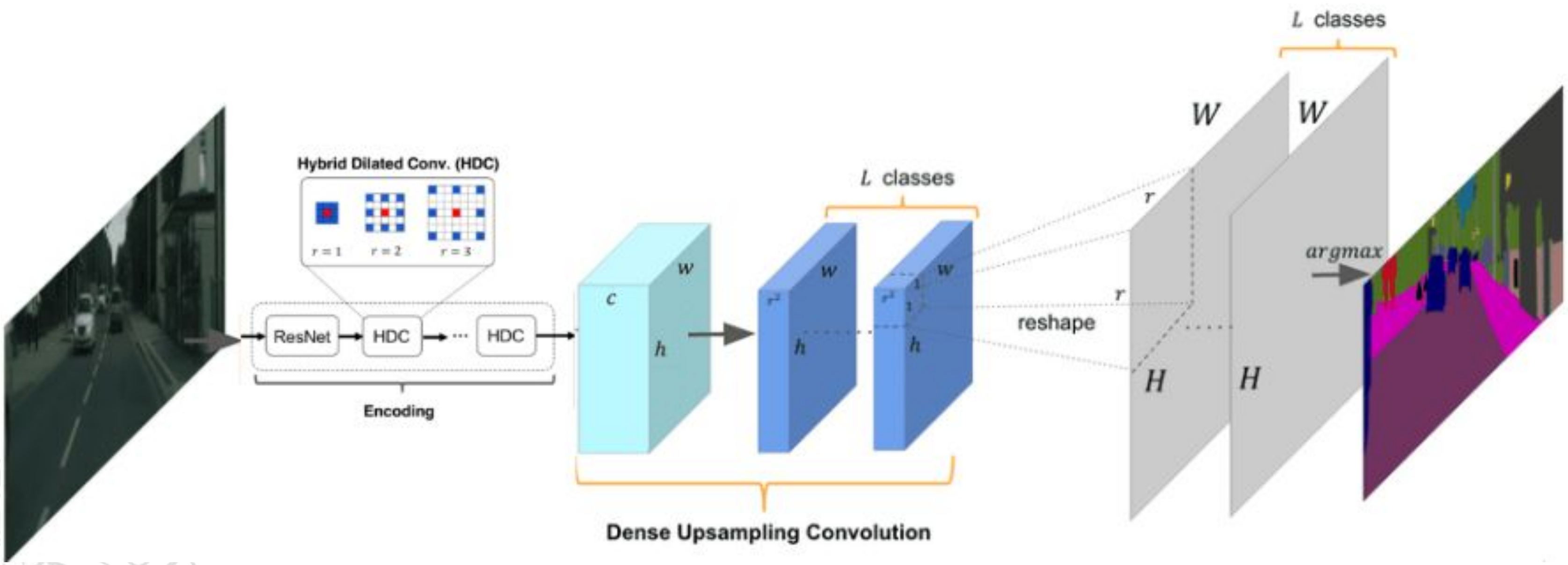
$$M_n = r_n, \qquad M_2 \leqslant K$$

DUC + HDC: Decoding

- Problem: Bilinear Upsampling

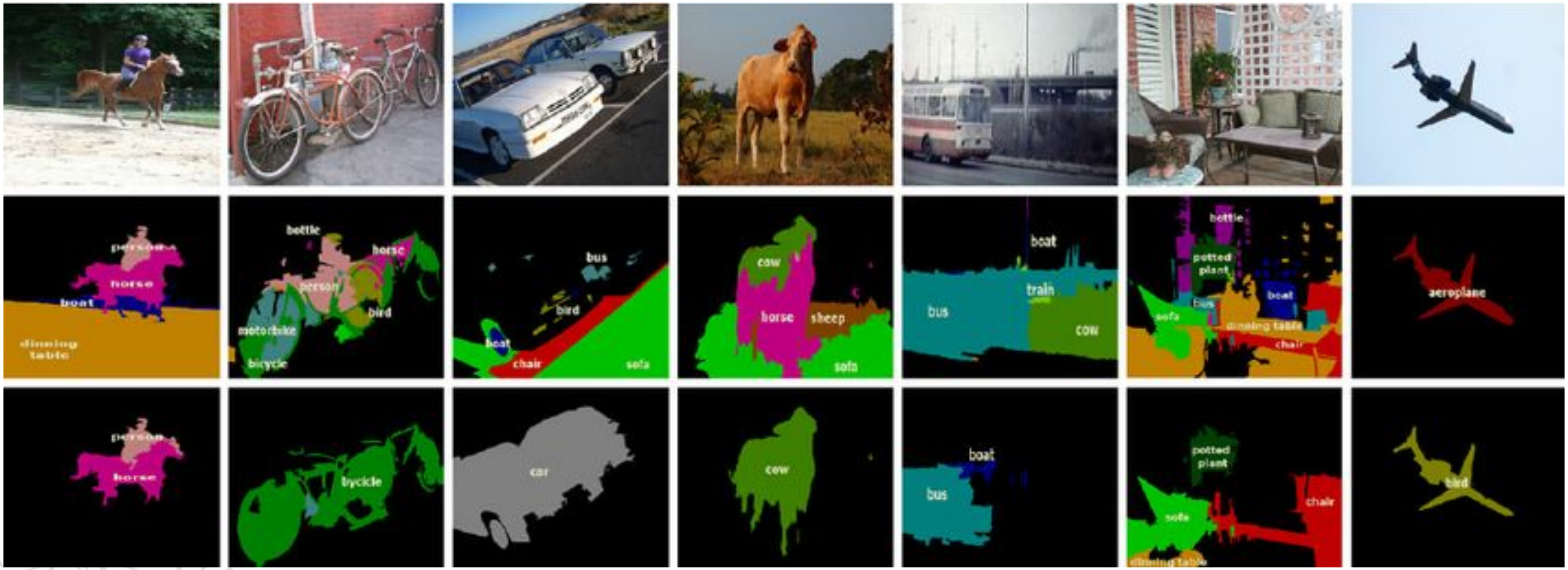


- Solution: Dense Upsampling Convolution (DUC)



PASCAL VOC2012

- Dataset: PASCAL VOC2012
  - 20 foreground object classes
  - 1 background class
  - 1456 training images, 1449 validation images and 1456 test images
  - Usually, the training set is augmented to have 10582 images



Cityscapes

- Dataset: Cityscapes
  - Street scenes from 50 different cities
  - High quality pixel-level annotations of 5 000 frame
  - 20 000 weakly annotated frames
  - 19 classes

DUC + HDC: Results

- Dataset: Cityscapes

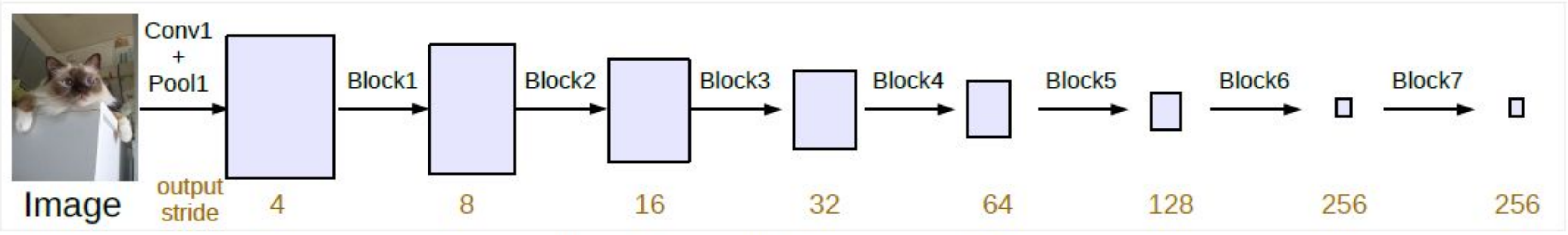
Method	mxNet	mIoU
Custom DUC+HDC	0.11.0	81.9
Original DUC+HDC	1.5.1	80.1

Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation

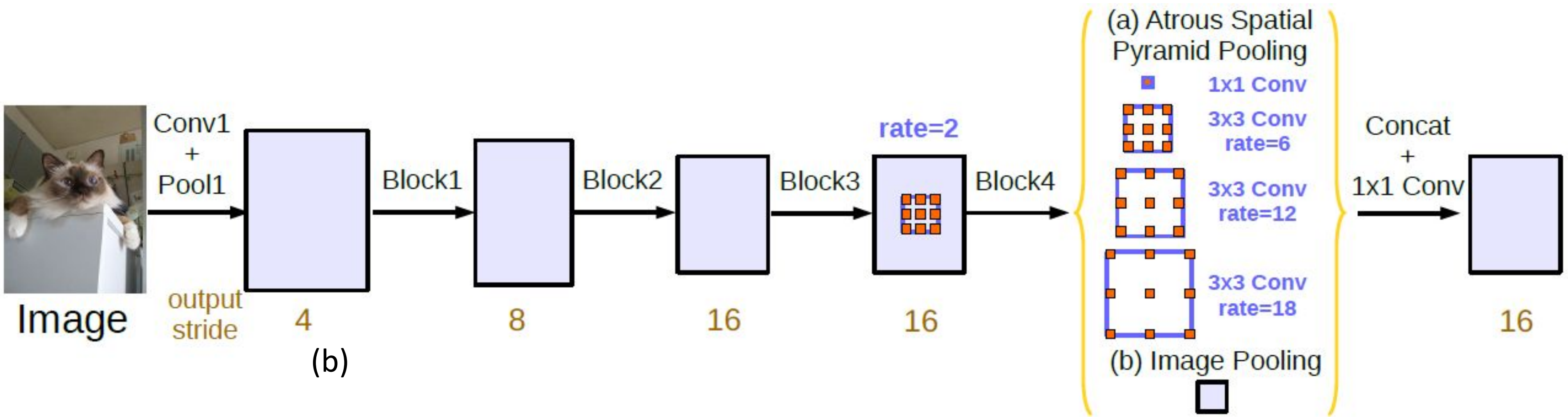
(Liang- Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam, 2018)

Deeplab V3+

- Deeplab V3
  - Going deeper without atrous convolution: location/spatial information is lost at the deeper layers.



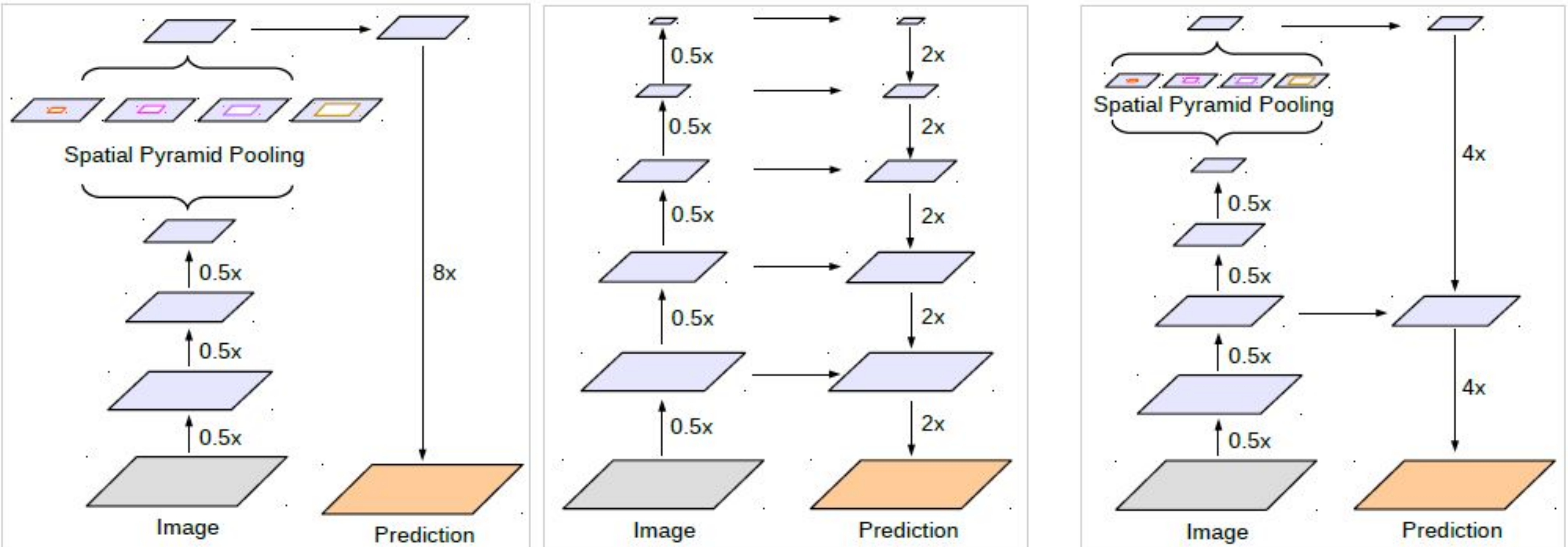
  - Going deeper with atrous convolution: information lost in decoding phase.



(b)

Deeplab v3+

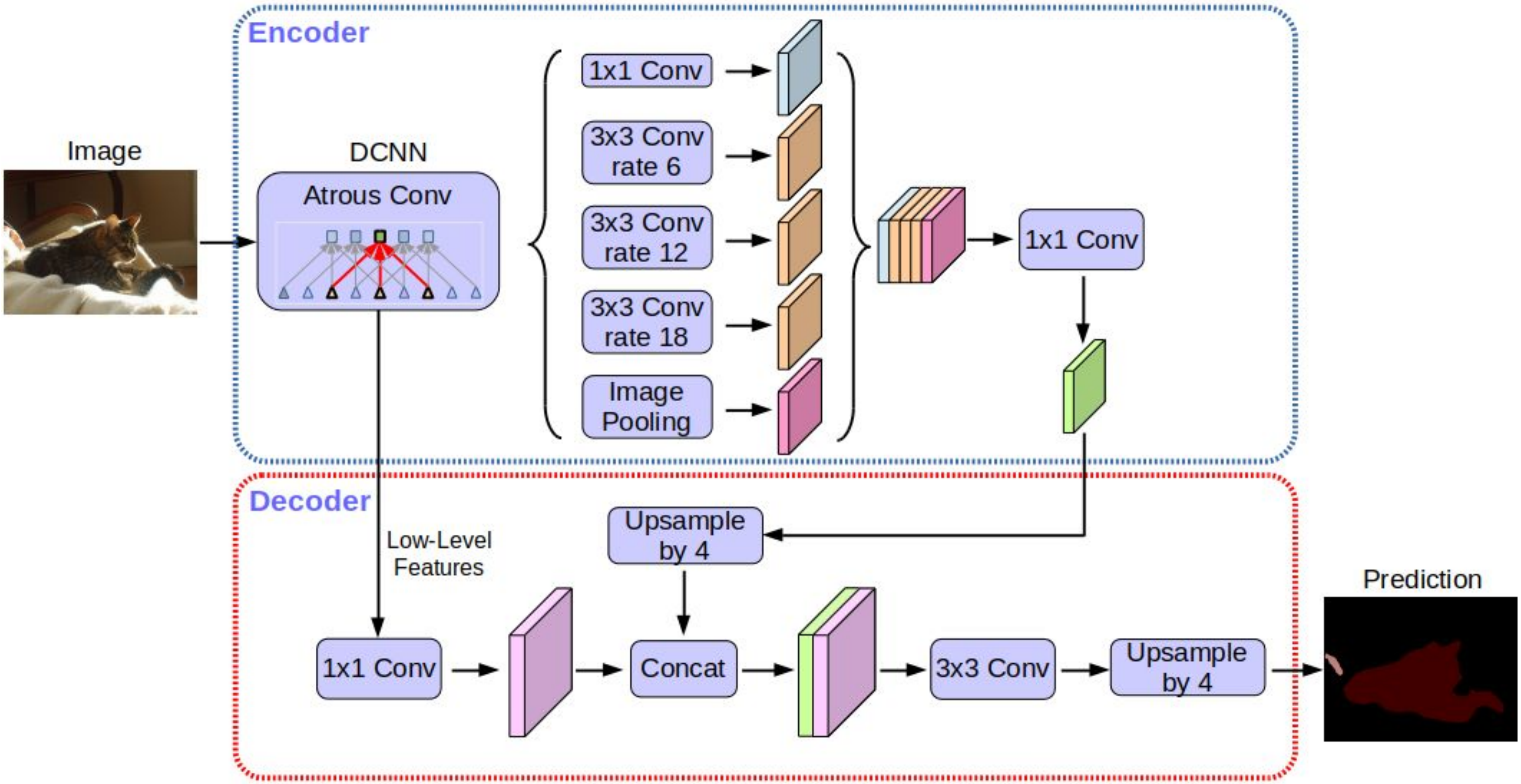
- Deeplab V3+



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Con

Deeplab v3+

- Deeplab V3+



Deeplab v3+: Results

- Dataset: Pascal VOC 2012

Method	Backbone	OS	mIoU
Custom Tensorflow	ResNet 101	16	77.31
Original Tensorflow	ResNet 101	16	78.85
Custom Pytorch	ResNet 101	16	78.31
Custom Pytorch	Mobilnet	16	70.89

- Dataset: Cityscapes

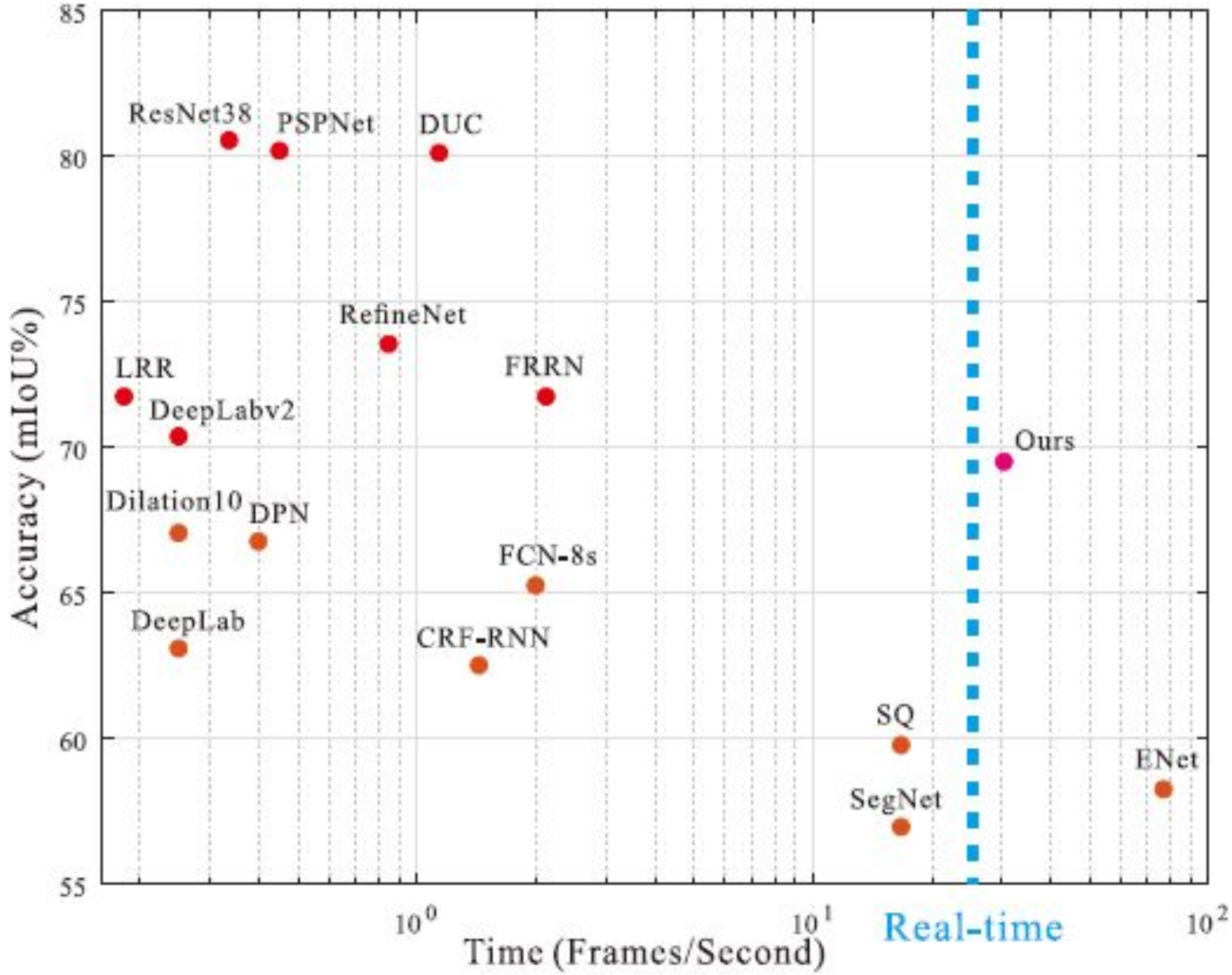
Method	Backbone	OS	mIoU
Custom Pytorch	Mobilnet	16	72.1
Original Tensorflow	ResNet 101	16	81.9

ICNet for Real-Time Semantic Segmentation on High-Resolution Images

(Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, Jiaya Jia, 2018)

ICNet

- Quality Vs Speed



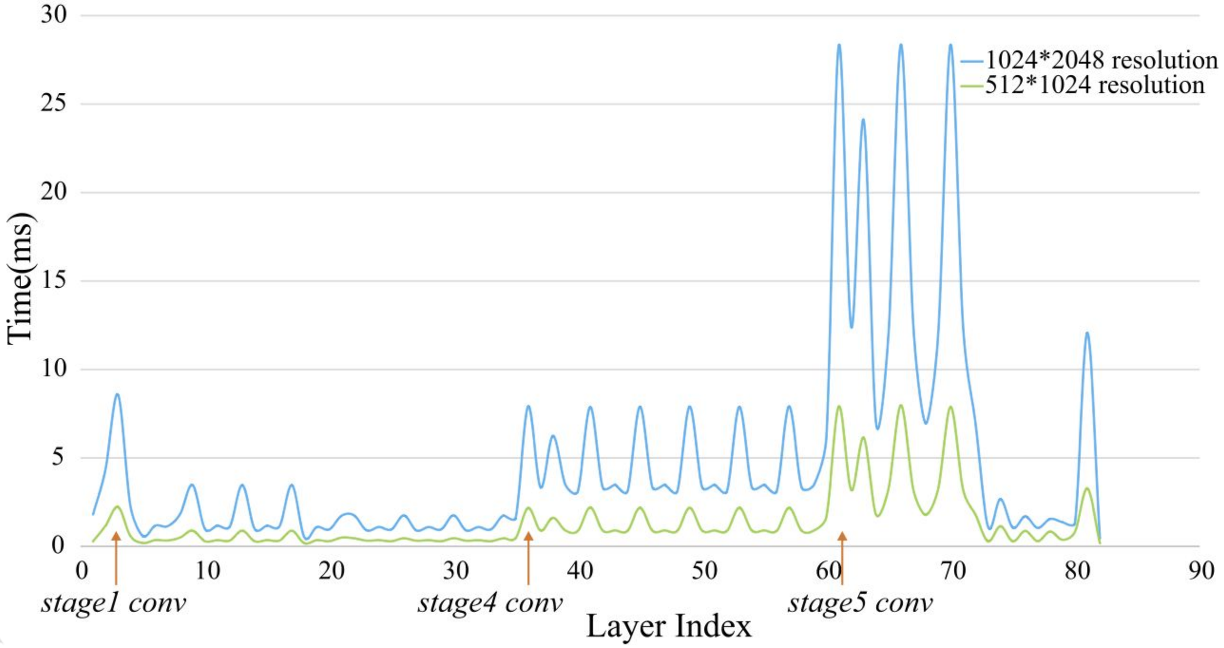
ICNet

- Convolution Speed Analysis (PSPNet50)

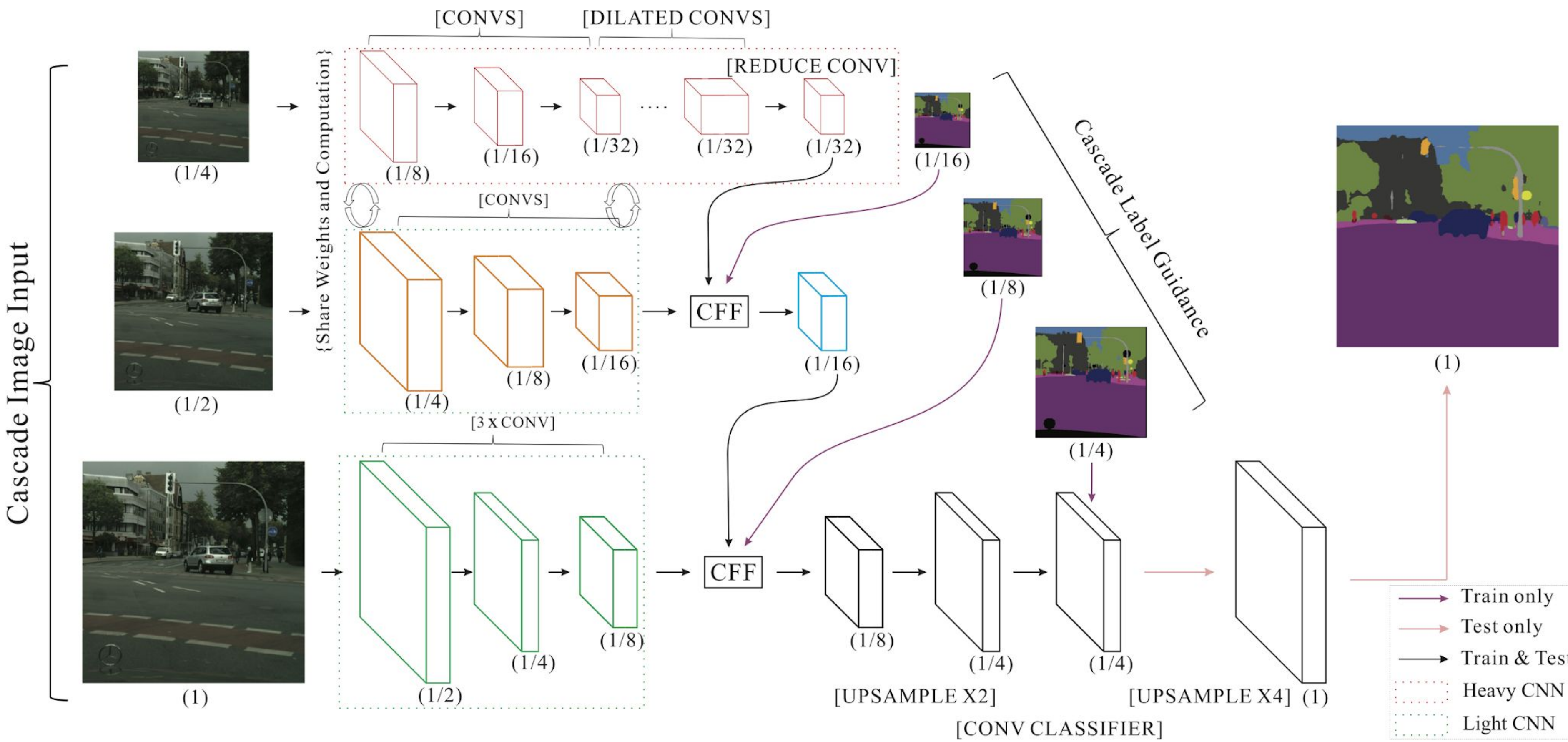
$$\Phi: V \longrightarrow U$$

$$V \in \mathbb{R}^{c \times h \times w} \quad U \in \mathbb{R}^{c' \times h' \times w'} \quad h'=h/s \quad w'=w/s$$

$$O(\Phi) \approx c'ck^2hw/s^2$$

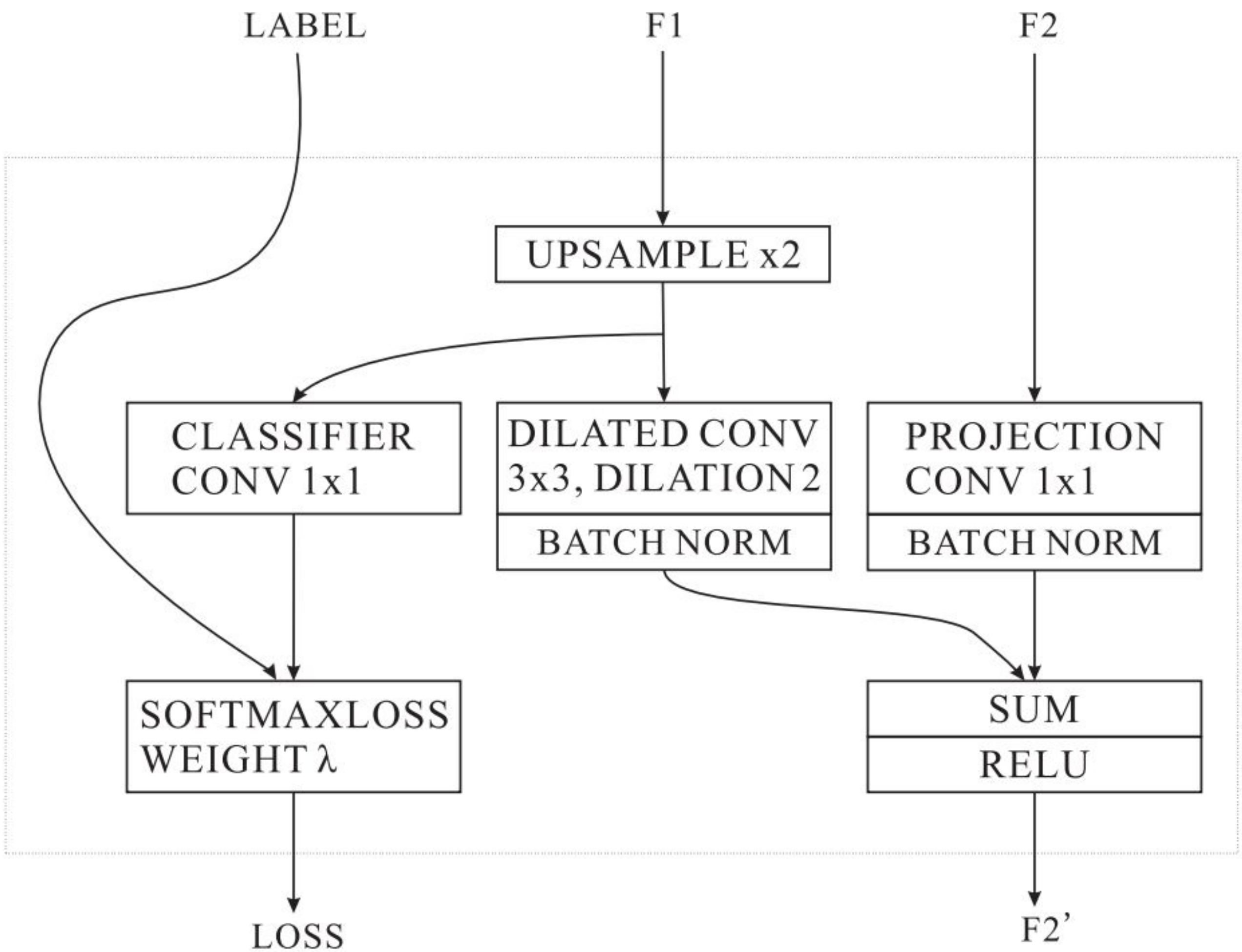


ICNet


[CONVS]
[DILATED CONVS]
[REDUCE CONV]
Cascade Image Input
(1/4)
(1/2)
(1)
{Share Weights and Computation}
(1/8)
(1/16)
(1/32)
(1/32)
(1/32)
[CONVS]
(1/4)
(1/8)
(1/16)
[3 x CONV]
(1/2)
(1/4)
(1/8)
CFF
CFF
(1/16)
(1/16)
(1/8)
(1/4)
Cascade Label Guidance
(1/8)
(1/4)
(1/4)
[UPSAMPLE X2]
[UPSAMPLE X4]
(1)
[CONV CLASSIFIER]
(1)
Train only
Test only
Train & Test
Heavy CNN
Light CNN

ICNet: CCF

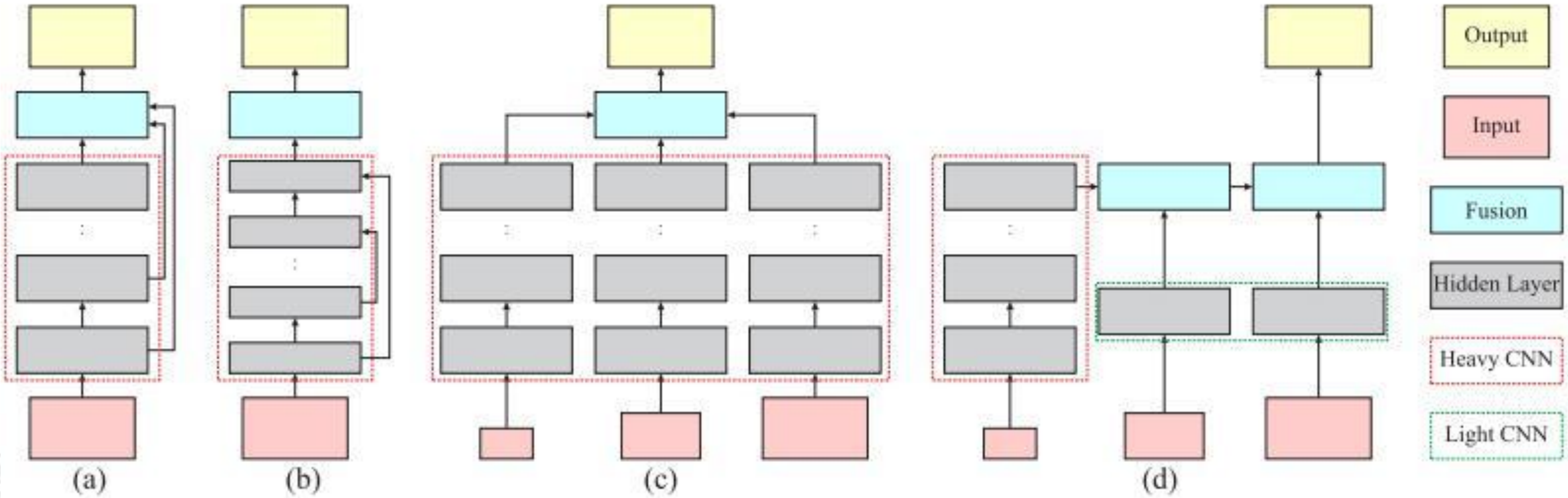
- Cascade Feature Fusion (CFF)



ICNet: CLG

- Cascade Label Guidance (CLG)

$$\mathcal{L} = -\sum_{t=1}^{\mathcal{T}} \lambda_t \frac{1}{\mathcal{Y}_t \mathcal{X}_t} \sum_{y=1}^{\mathcal{Y}_t} \sum_{x=1}^{\mathcal{X}_t} \log \frac{e^{\mathcal{F}^t_{\hat{n},y,x}}}{\sum_{n=1}^{\mathcal{N}} e^{\mathcal{F}^t_{n,y,x}}}$$



ICNet: Results

- Dataset: Cityscapes
  - Experiments on gtx 1070 max-q

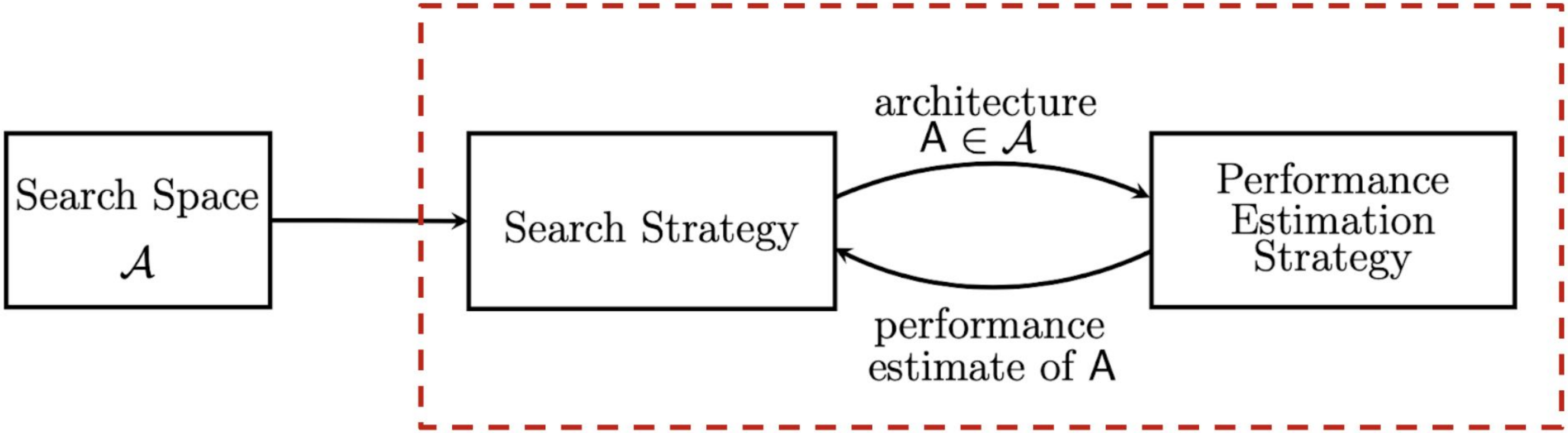
Method	Model	DR	mIoU	Time(ms)	Frame(fps)
ICNetC	30k_bn	no	67,35	66	15,14
ICNetC	90k_bn	no	80,08	69	14,33

  - Experiments on TitanX Maxwell

Method	Model	DR	mIoU	Time(ms)	Frame(fps)
ICNetO	30k	no	67,7	33	30,3
ICNetO	90k	no	70,6	33	30,3

NAS

One-shot approach:
learning model architecture parameters and weights together

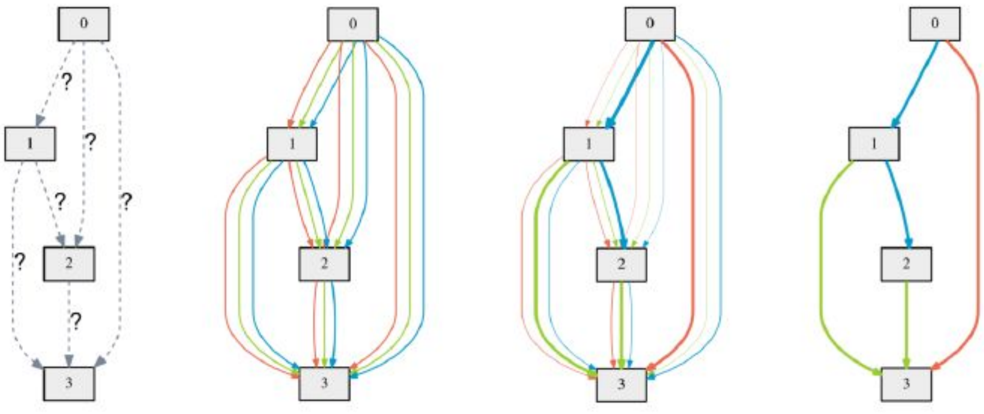


DARTS: Differentiable Architecture Search

(Hanxiao Liu, Karen Simonyan, Yiming Yang 2019)

DARTS: Search Space

- Cell Architecture



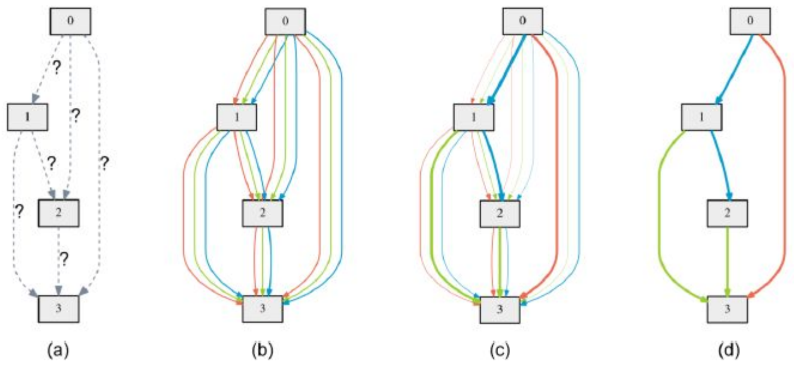
(a) (b) (c) (d)

- 3×3 depthwise-separable conv
- 5×5 depthwise-separable conv
- 3×3 atrous conv with rate 2
- 5×5 atrous conv with rate 2
- 3×3 average pooling
- 3×3 max pooling
- skip connection
- no connection (zero)

Each intermediate node is computed based on all of its predecessors:

$$x^{(j)} = \sum_{i<j} o^{(i,j)}\left(x^{(i)}\right)$$

DARTS: Continuous Relaxation and Optimization



Softmax over all possible operations:

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

A discrete architecture can be obtained by replacing each mixed operation o ̅(i,j) with the most likely operation:

$$o^{(i,j)} = \mathrm{argmax}_{o \in \mathcal{O}} \; \alpha_o^{(i,j)}$$

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$

$$\text{s.t.} \quad w^*(\alpha) = \mathrm{argmin}_w \; \mathcal{L}_{train}(w, \alpha)$$

DARTS: Approximate Architecture Gradient

- First Order Approximation

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$
$$\text{s.t.} \quad w^*(\alpha) = \text{argmin}_w \; \mathcal{L}_{train}(w, \alpha)$$

$$\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha) \approx \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha) \quad (1)$$

Algorithm 1: DARTS – Differentiable Architecture Search

Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge (i, j)
while *not converged* **do**
  1. Update weights w by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$
  2. Update architecture α by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$

Replace $\bar{o}^{(i,j)}$ with $o^{(i,j)} = \text{argmax}_{o \in \mathcal{O}} \, \alpha_o^{(i,j)}$ for each edge (i, j)

DARTS: Approximate Architecture Gradient

- ## Second Order Approximation

Applying chain rule to the approximate architecture gradient (eq. 1) yields:

$$\nabla_\alpha \mathcal{L}_{val}(w', \alpha) - \xi \nabla^2_{\alpha,w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \qquad (2)$$

where: $w' = w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha)$

Let be ε a small scalar and $w^{\pm} = w \pm \epsilon \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$

Then:

$$\nabla^2_{\alpha,w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \approx \frac{\nabla_\alpha \mathcal{L}_{train}(w^+, \alpha) - \nabla_\alpha \mathcal{L}_{train}(w^-, \alpha)}{2\epsilon}$$

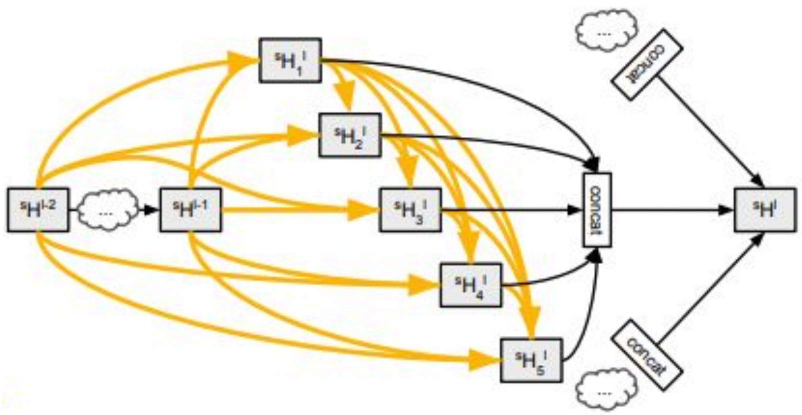
$O(|\alpha||w|)$ to $O(|\alpha| + |w|)$

Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation

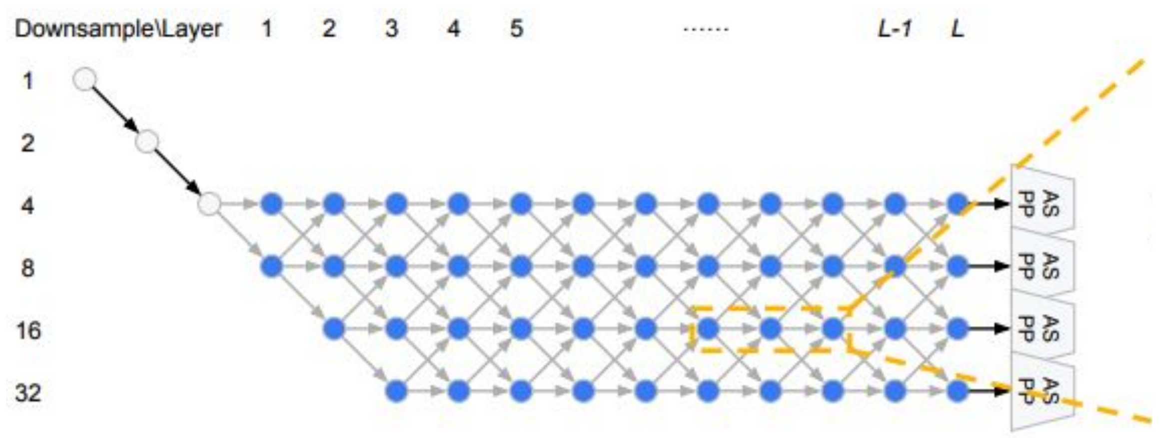
(Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, Li Fei-Fei
Johns Hopkins University - Google - Stanford University)

Auto-DeepLab: Network Level Search Space

- Cell Architecture:

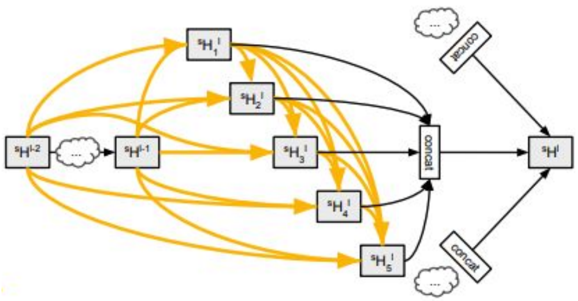


- Network Architecture:



Auto-DeepLab: Cell Architecture

- Cell Architecture

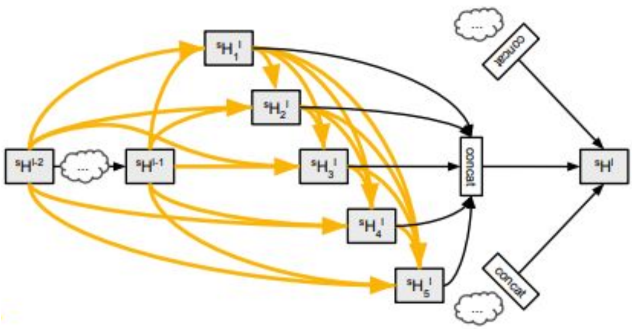


- a *cell* is a directed acyclic graph consisting of *B* blocks.
- a *block i* in cell *l* may be specified using a 5-tuple (I_1, I_2, O_1, O_2, C), where $I_1, I_2 \in I_i^l$, $O_1, O_2 \in O$ and $C \in C$.
- The cell's output tensor H_l is simply the concatenation of the blocks' output tensors $H_1^l, \ldots, H_B^l$ in this order.

- The set of possible input tensors, I_i^l, consists of the output of the previous cell H^{l-1}, the output of the previous-previous cell H^{l-2}, and previous blocks' output in the current cell $\{H_1^l, \ldots, H_{i-1}^l\}$.

Auto-DeepLab: Cell Architecture

- Cell Architecture



Every block's output tensor depends on:

i. $$H_i^l = \sum_{H_j^l \in \mathcal{I}_i^l} O_{j\to i}(H_j^l)$$

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

where $$\sum_{k=1}^{|\mathcal{O}|} \alpha_{j\to i}^k = 1 \qquad \forall i, j$$

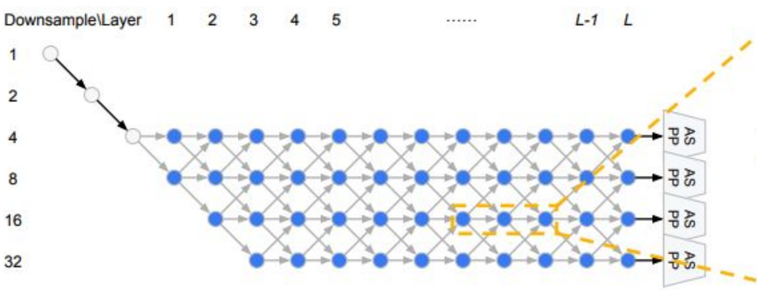
$$\alpha_{j\to i}^k \geq 0 \qquad \forall i, j, k$$

Together with Eq. (i) and Eq. (ii), the cell level update may be summarized as:

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

Auto-DeepLab: Network Architecture

- Network Architecture



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

- The network level update is:

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

where s = 4, 8, 16, 32 and l = 1, 2, . . . , L.

- The scalars β are normalized such that:

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

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

Auto-DeepLab: Optimization

- Optimization
  - Apply first-order approximation and partition the training data into two disjoint sets *trainA* and *trainB*.
  - The optimization alternates between:

1. Update network weights w by $\nabla_w \mathcal{L}_{trainA}(w, \alpha, \beta)$
2. Update architecture α, β by $\nabla_{\alpha,\beta} \mathcal{L}_{trainB}(w, \alpha, \beta)$

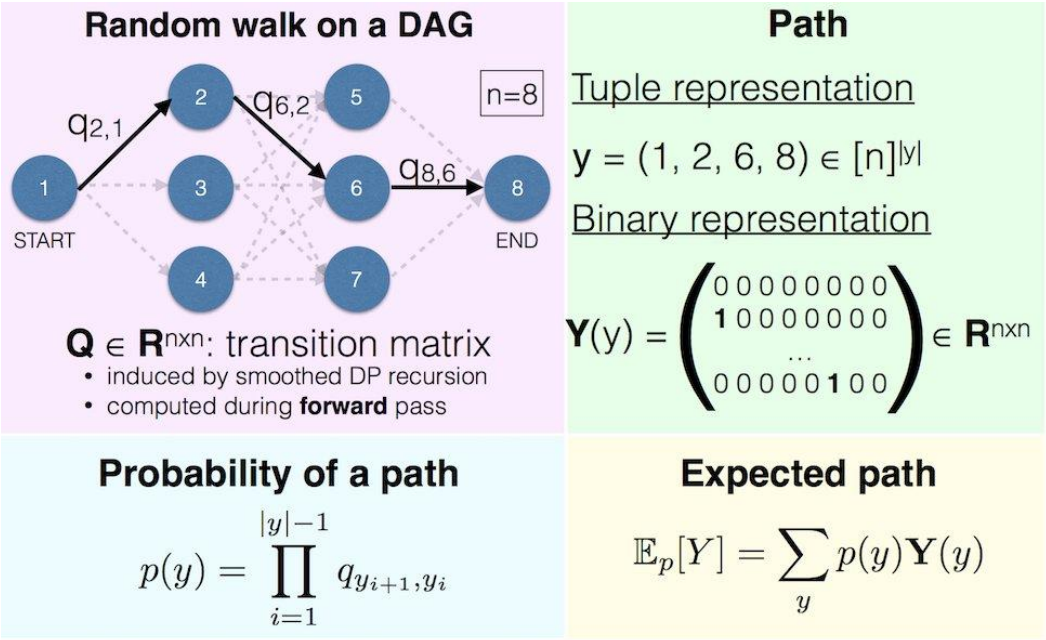
where the loss function *L* is the cross entropy calculated on the semantic segmentation mini-batch.

- Decoding Discrete Architecture
  - Cell architecture

Decoding by first retaining the 2 strongest predecessors for each block:

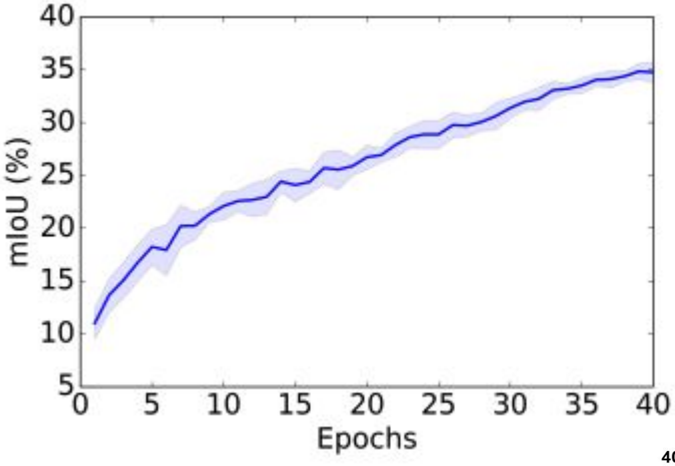
$$\max_{k, O^k \neq zero} \alpha^k_{j \to i}$$

  - Network Architecture

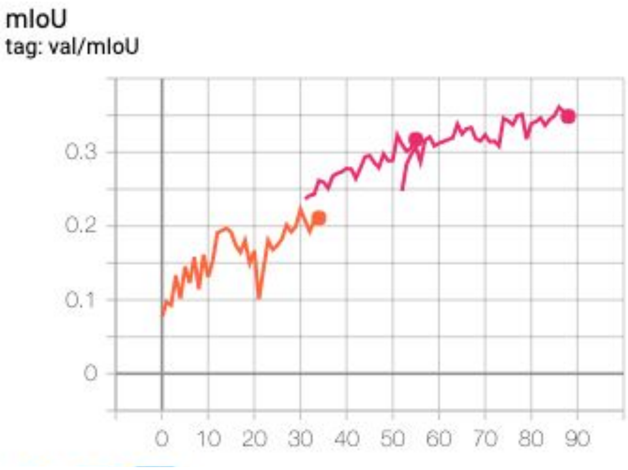


Auto-DeepLab: Results

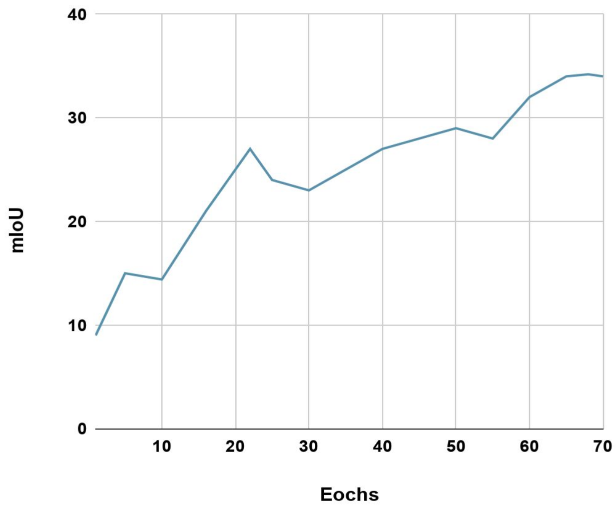
- Search results



filter_multiplier 8, crop_size 512



filter_multiplier 4, crop_size 224



filter_multiplier 8, crop_size 224

Auto-DeepLab: Results

- Retrain results

  - Cityscapes

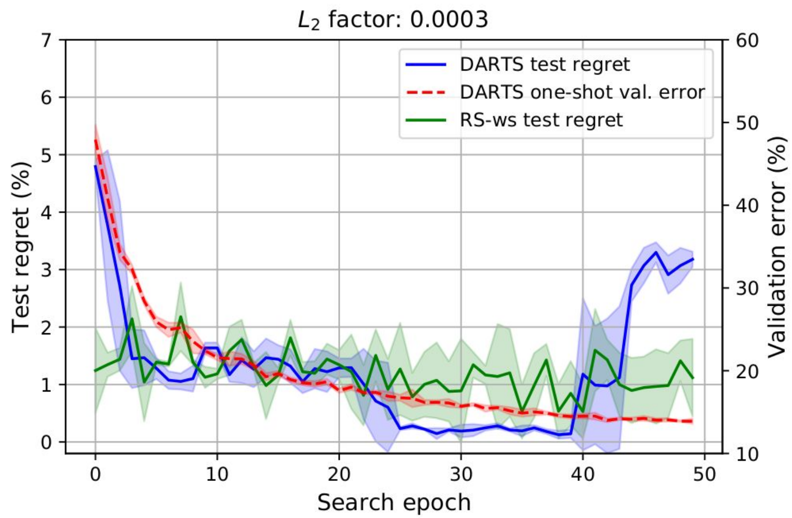
Method	***F***	***Crop-Size***	**mIoU(%)**
Auto-DeepLab-Half	32	256	69.41
Auto-DeepLab-S	20	512	79.74
Auto-DeepLab-M	32	512	80.04
Auto-DeepLab-L	48	512	80.33

  - Multi-Scale + Coarse

Auto-DeepLab-L	48	512	82.1

Conclusions

- Understanding and Robustifying Differentiable Architecture Search (Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox & Frank Hutter, ICLR 2020)



- a simple heuristic:

Let $\underline{\lambda}^{\alpha}_{max}(i)$ denote the value of λ^{α}_{max} smoothed over k = 5 epochs around i;

then, we stop if $\lambda^{\alpha}_{max}(i - k)/\lambda^{\alpha}_{max} < 0.75$ and return the architecture from epoch i − k.

Conclusions

- Results comparison
  - Cityscapes

Method	ImageNet	mIoU(%)
HDC+DUC	✓	81.9 (+1,8)
DeepLabv3+	✓	81.9 (-9,8)
ICNet	✓	80.08 (+9,48)
AutoDeepLab		82.1 (-12,69)

Conclusions

Thanks for the attention!