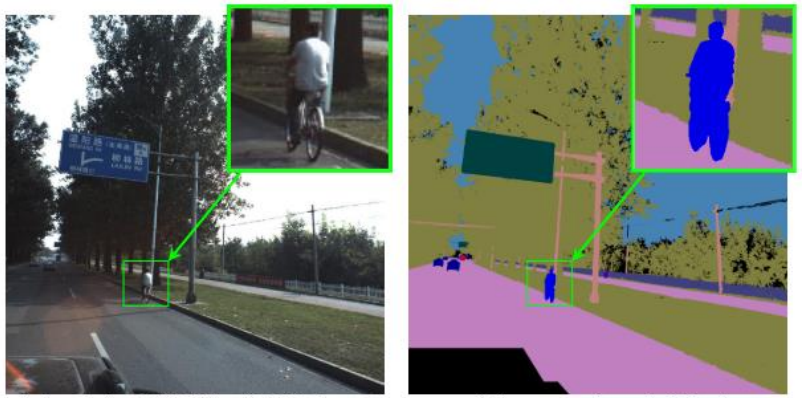
文献分享——分割相关

陈炜祥

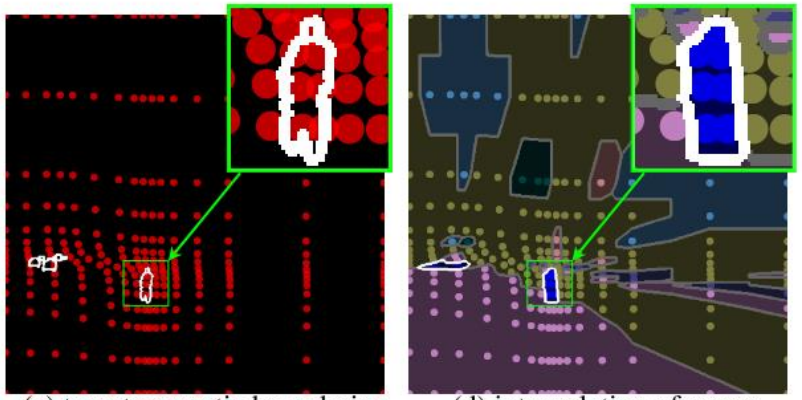
2020-04-16

Efficient Segmentation: Learning Downsampling Near Semantic Boundaries

- Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, Yuri Boykov. **Efficient Segmentation: Learning Downsampling Near Semantic Boundaries.** *In International Conference on Computer Vision (ICCV)*, 2019
- 提供代码：https://github.com/dmitrii-marin/adaptive-sampling
- 动机：
  - 在深度学习进行分割时，由于内存原因不可避免地需要进行降采样
  - 降采样带来的边缘模糊化，将引入O(D √ N)的误差
  - 不均匀的采样可以减少物体边缘的模糊化
  - 同时引入的不均匀采样有助于提高原本尺度较小物体在降采样之后的尺度占比



(a) original 2710×2710 image (b) ground truth labels



(c) target semantic boundaries and adaptive sampling locations (d) interpolation of sparse classifications (target is white)

Prior work

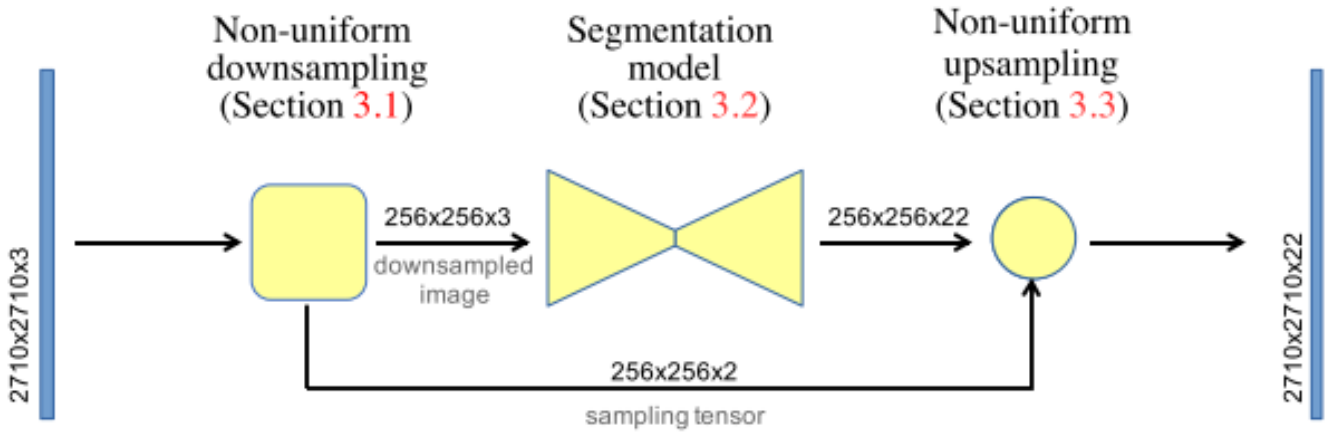
	two-stage [18, 19, 21]	ours Sec. 3	single-stage [6, 35, 44]
accuracy	++	+	-
speed	-	+	++
multi-object speed	- -	+	++
simplicity	-	+	++
multi-scale	++	+	-
boundary precision	++	+	-

- **Classification based networks**：
  - FCN、dilated convolutions.
  - **classification models** tend to have many features in the deeper layers. That results in an **extensive resources consumption** when increasing the resolution of later feature maps
- **Hourglass models:**
  - First produce low resolution deep features and then gradually upsample the features employing common network operations and skip connections
- **Multi-scaled models**:
  - two-stage approaches

Prior work

- **Spatial Transformer Networks:** learn spatial transformations (warping) of the CNN input.
  - They do not use their approach in the context of pixel-level predictions (e.g. segmentation) and do not consider the **inverse transformations**
- **Deformable convolutions:** augment the spatial sampling locations with additional adaptive offsets in **last few layers**
  - Our approach focuses on choosing the best locations in the original image and thus has access to more information

Boundary Driven Adaptive Downsampling



- 数学符号:

$I = \{I_{ij}\} \in R^{H\times W\times C}$，将每个像素$I[u,v]$记作相对位置$(u,v) \in [0,1]^2$

变形向量 $\varphi \in [0,1]^{h\times w\times 2}$的每一个元素$(\varphi_{ij}^0, \varphi_{ij}^1)$是像素点$ij$处的对应采样点位置

因此采样运算可以： $R^{H\times W\times C} \rightarrow R^{h\times w\times C}$

- 根据目标设计损失函数:
  - 降采样后边界依旧清晰
  - 采样场平滑
- $b(u_{ij})$ 是离u_{ij}最近一个分割边界的坐标值

$$E(\phi) = \sum_{i,j} \|\phi_{ij} - b(u_{ij})\|^2 + \lambda \sum_{\substack{|i-i'|+\\ |j-j'|=1}} \|\phi_{ij} - \phi_{i'j'}\|^2$$

$$\phi_{1\,j}^0 = 0 \quad \& \quad \phi_{hj}^0 = 1, \quad 1 \le j \le w,$$
$$\phi_{i\,1}^1 = 0 \quad \& \quad \phi_{iw}^1 = 1, \quad 1 \le i \le h.$$

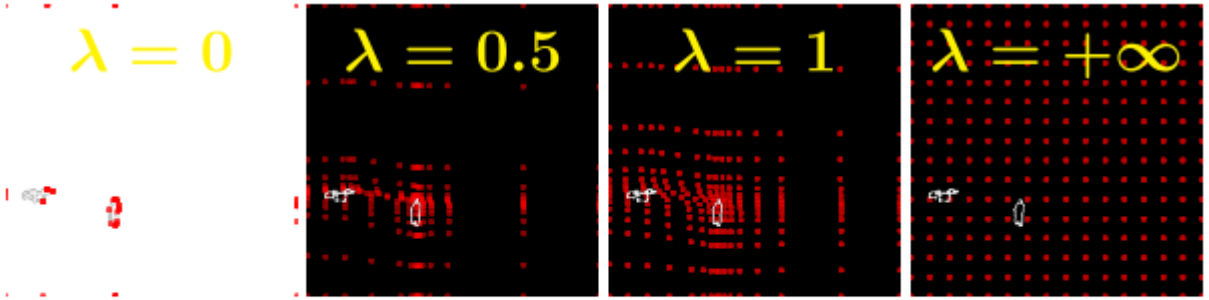


Figure 3: Boundary driven sampling for different λ in (2). Extreme λ sample either semantic boundaries (left) or uniformly (right). Middle-range λ yield in-between sampling.

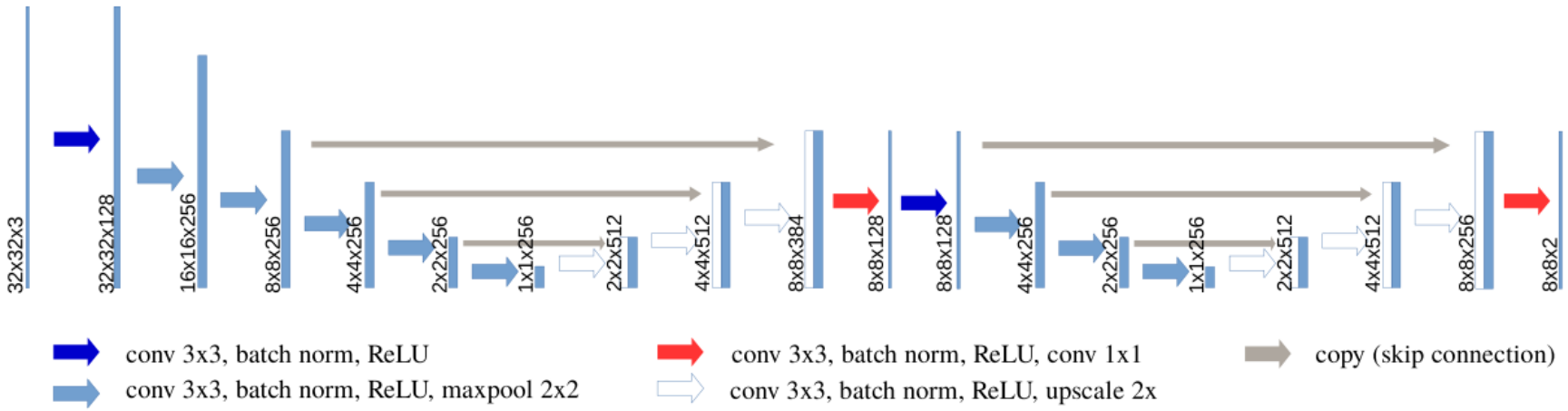


Figure 5: Double U-Net model for predicting sampling parameters. The depth of the first sub-network can vary (depending on the input resolution). The structure of the second sub-network is kept fixed. To improve efficiency, we use only one convolution (instead of two in [44]) in each block. The number of features is 256 in all layers except the first and the last one. We also use padded convolutions to avoid shrinking of feature maps, and we add batch normalization after each convolution.

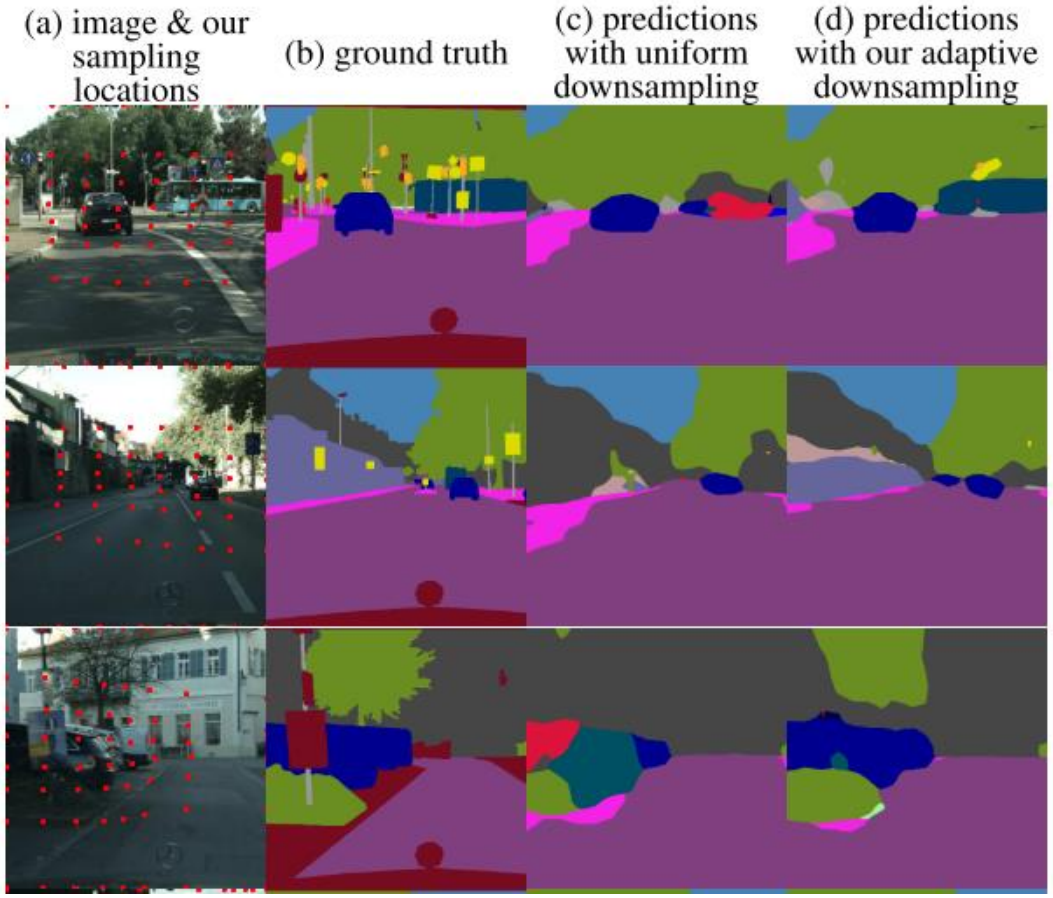
- A relatively small **auxiliary network** to predict the sampling tensor without boundaries: two U-Net
- The motivation for stacking sub-networks is to model the **sequential processes** of boundary computation and **sampling points selection**

Segmentation Model & Upsampling

- Segmentation: Multiple models (U-Net, PSP-Net and Deeplabv3+)
- Upsampling:
  - 由于加入了covering constraints，所以采样前后的图像覆盖的区域是一致的，能够直接插值复原
  - 先将采样点Delaunay triangulation，然后barycentric interpolation
  - 加速算法Bresenham’s algorithm

Experiments

- 数据集:
  - ApolloScape：105K training and 8K validation images of size 3384×2710
  - CityScapes：5K annotated images of size 1024× 2048 with 19 classes in evaluation
  - Synthia：13K HD images taken from an array of cameras moving randomly through a city
  - Person segmentation：5711 high-resolution images with 6884 high-quality annotated person instances

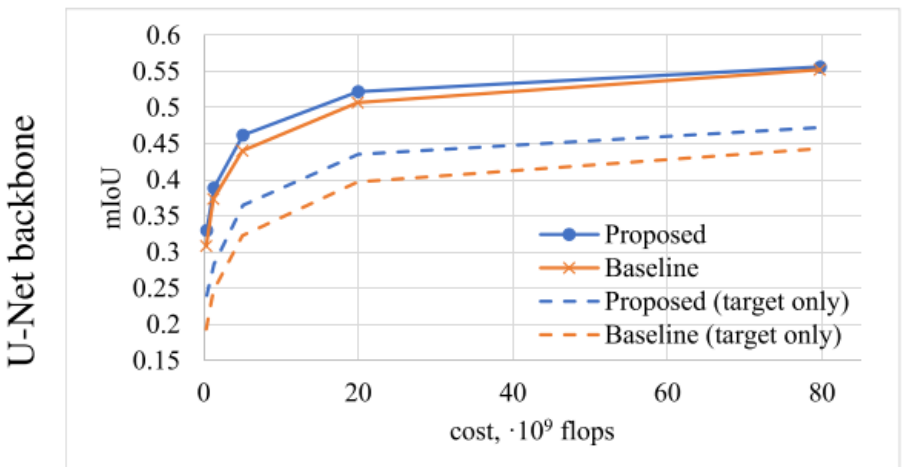


	downsample resolution	flops, $\cdot 10^9$	non-target classes, IoU														**target classes**, IoU								mIoU	
			road	sidewalk	traffic cone	road pile	fence	traffic light	pole	traffic sign	wall	dustbin	billboard	building	vegatation	sky	**car**	**motor-bicycle**	**bicycle**	**person**	**rider**	**truck**	**bus**	**tricycle**	all classes	**target classes**
Ours	32	0.38	**0.92**	**0.38**	**0.17**	**0.00**	**0.49**	0.11	0.08	0.44	**0.28**	**0.03**	**0.00**	0.74	0.86	0.84	**0.66**	**0.07**	**0.27**	**0.02**	**0.03**	**0.34**	**0.52**	**0.01**	**0.24**	**0.24**
Baseline	32	**0.31**	0.92	0.29	0.13	0.00	0.43	**0.14**	**0.11**	**0.53**	0.18	0.00	0.00	**0.74**	**0.87**	**0.89**	0.59	0.04	0.26	0.01	0.02	0.20	0.44	0.00	0.19	0.19
Ours	64	1.31	0.94	0.39	**0.31**	**0.02**	**0.56**	0.25	0.17	0.61	**0.41**	**0.08**	**0.00**	0.78	0.89	0.87	**0.76**	**0.10**	**0.33**	**0.04**	**0.03**	**0.44**	**0.53**	**0.04**	**0.28**	**0.28**
Baseline	64	**1.24**	**0.94**	**0.40**	0.30	0.01	0.52	**0.30**	**0.22**	**0.64**	0.29	0.04	0.00	**0.79**	**0.90**	**0.91**	0.70	0.06	0.31	0.02	0.03	0.32	0.52	0.03	0.25	0.25
Ours	128	5.05	0.95	**0.51**	**0.43**	**0.07**	**0.61**	0.44	0.29	0.71	**0.47**	**0.13**	**0.01**	0.82	0.91	0.88	**0.83**	**0.16**	**0.41**	**0.08**	**0.05**	**0.57**	**0.76**	0.06	**0.36**	**0.36**
Baseline	128	**4.98**	**0.96**	0.39	0.43	0.05	0.59	**0.45**	**0.36**	**0.73**	0.37	0.11	0.00	**0.83**	**0.92**	**0.93**	0.80	0.10	0.38	0.06	0.03	0.44	0.70	**0.06**	0.32	0.32
Ours	256	19.99	0.96	0.44	**0.51**	**0.13**	**0.66**	**0.58**	0.42	0.78	**0.58**	**0.27**	**0.00**	0.84	0.92	0.89	**0.88**	**0.21**	**0.47**	**0.18**	0.04	**0.65**	0.80	**0.24**	**0.44**	**0.44**
Baseline	256	**19.92**	**0.97**	**0.48**	0.49	0.13	0.64	0.58	**0.46**	**0.79**	0.48	0.24	0.00	**0.85**	**0.94**	**0.94**	0.86	0.17	0.42	0.15	**0.04**	0.60	**0.83**	0.10	0.40	0.40
Ours	512	79.76	0.97	0.44	0.54	**0.21**	0.68	0.63	0.49	0.80	**0.67**	0.36	0.00	0.85	0.93	0.90	**0.91**	**0.24**	**0.52**	**0.30**	**0.06**	**0.75**	0.81	**0.19**	**0.47**	**0.47**
Baseline	512	**79.68**	**0.97**	**0.47**	**0.55**	0.20	**0.68**	**0.67**	**0.54**	**0.83**	0.59	**0.36**	**0.00**	**0.87**	**0.94**	**0.94**	0.90	0.21	0.49	0.26	0.03	0.68	**0.84**	0.13	0.44	0.44

Table 2: Per class results on the validation set of ApolloScape. Our adaptive sampling improves overall quality of segmentation. Target classes (bold font on the top row) consistently benefit for all resolutions.

Achieves a mIoU gain of 3-5% for target classes and up to 2% overall.

Produces better results even under fixed computational budgets



	downsample resolution	auxiliary net resolution	flops, $\cdot 10^9$	mIoU	downsample resolution	auxiliary net resolution	flops, $\cdot 10^9$	mIoU
backbone	PSP-Net [57]				Deeplabv3+ [8]			
ours	64	32	4.37	**0.32**	160	32	17.54	**0.58**
baseline		-	4.20	0.29		-	17.23	0.54
ours	128	32	11.25	**0.43**	192	32	25.12	**0.62**
baseline		-	11.08	0.40		-	24.81	0.61
ours	256	32	44.22	**0.54**	224	32	34.08	**0.65**
baseline		-	44.05	0.54		-	33.77	0.62

Table 3: CityScapes results with different backbones.

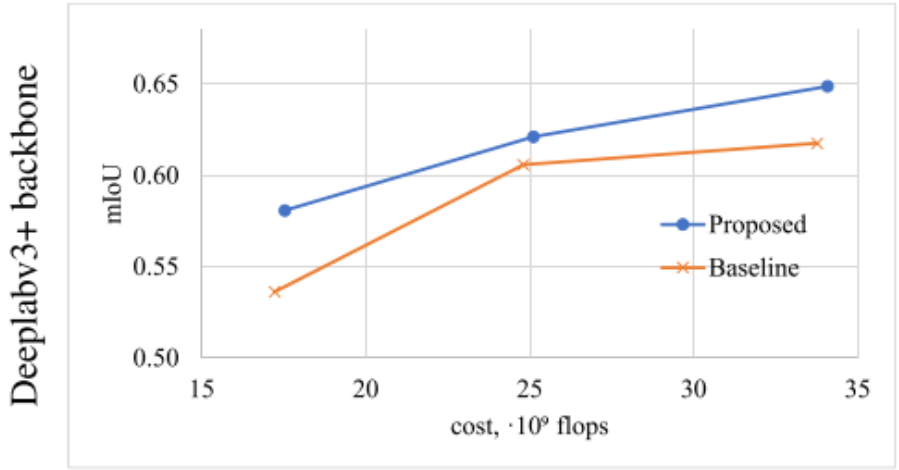


Figure 8: Cost-performance analysis on **CityScapes** with PSP-Net and Deeplabv3+ baselines for varying downsampling size, see Tab. 3. Our content-adaptive downsampling gives better results with the same computational cost.

	downsample resolution	flops, $\cdot 10^9$	all classes	target classes
ours	32	0.38	**0.67**	**0.61**
baseline		0.31	0.65	0.58
ours	64	1.40	**0.77**	**0.73**
baseline		1.23	0.76	0.71
ours	128	5.49	**0.86**	**0.83**
baseline		4.93	0.84	0.81
ours	256	21.85	**0.92**	**0.91**
baseline		19.74	0.91	0.89

Table 4: Synthia results (mIoU). With the same input resolution our approach improves the segmentation quality.

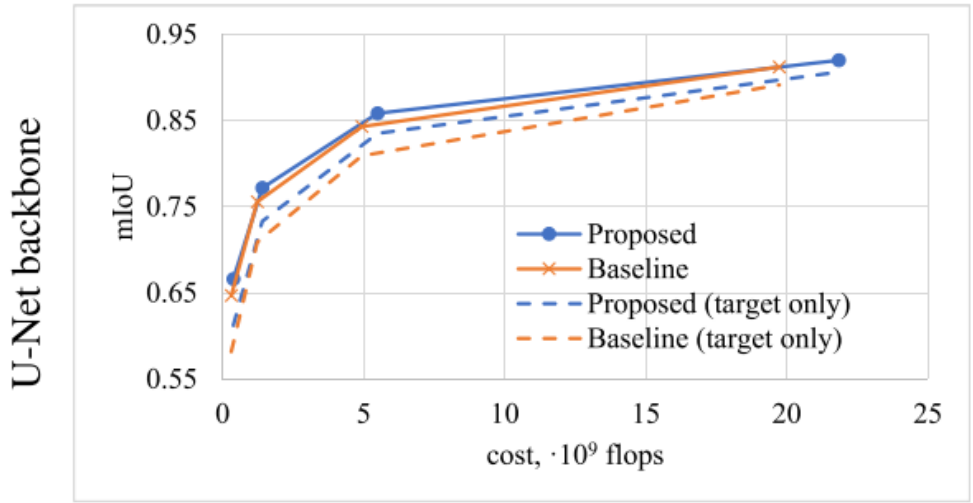
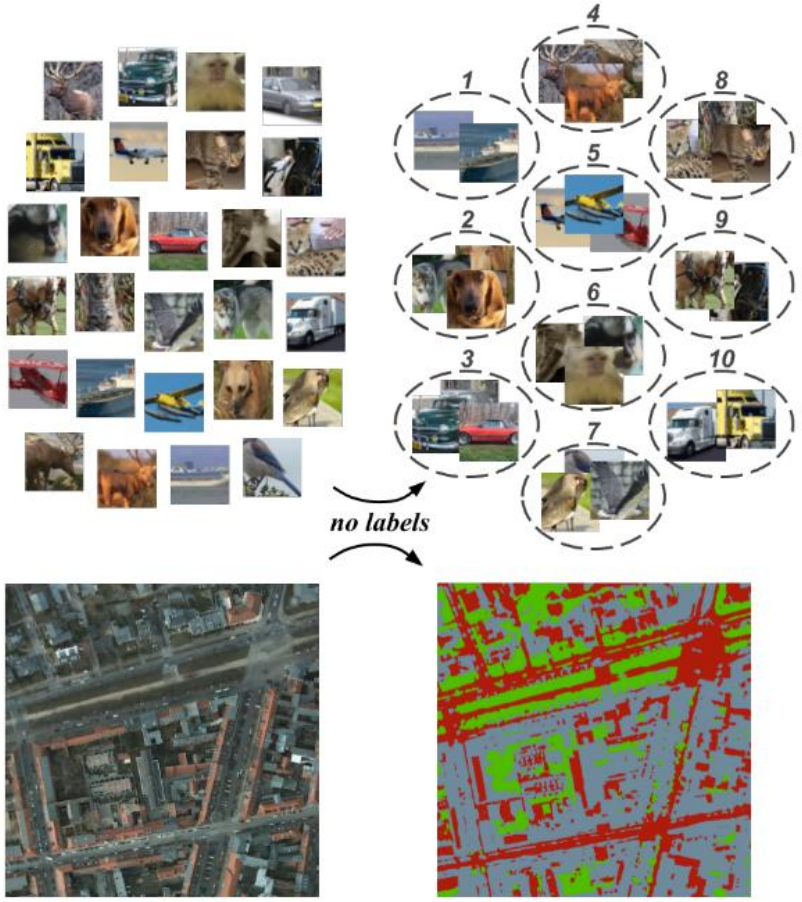


Figure 9: Cost-performance analysis on **Synthia** dataset. Our approach performs better for target classes (with a tie on all classes).

Invariant Information Clustering for Unsupervised Image Classification and Segmentation

- Ji, Xu & Vedaldi, Andrea & Henriques, Joao. (2019). Invariant Information Clustering for Unsupervised Image Classification and Segmentation. 9864-9873. 10.1109/ICCV.2019.00996.
- 源码：https://github.com/xu-ji/IIC
- 动机：
  - 用无监督的方式训练分类或者分割网络
- 相关工作：
  - 无监督聚类方案：传统方案
  - 聚类+深度学习：
    - 用information做监督：IMSAT、DeepINFOMAX
    - intermediate representation learning：DeepCluster
    - 容易出现degeneracy或者流程复杂
  - IIC is a generic clustering algorithm that directly trains a randomly initialised neural network into a classification function, end-to-end and without any labels



Invariant Information Clustering

- 数学定义:
  - x, x'是两张同类型的不同图片，记其联合概率为$P(x, x')$
  - 记一个IIC变化为$\Phi: X \rightarrow Y$，要求保留x, x'的共有信息 $\max_{\Phi} I(\Phi(\mathbf{x}), \Phi(\mathbf{x}')),$
  - 用一个深度网络的bottleneck表示Φ，且$\Phi(x) \in [0,1]^C$，其中C是聚类的类别数目，这里设置成已知的分类数目，等价于将Φ 定义为$P(z = c|x) = \Phi_c(x)$
  - 这个定义下联合概率变为 $\mathbf{P} = \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}'_i)^\top.$
  - 此时的目标函数为 $I(z, z') = I(\mathbf{P}) = \sum_{c=1}^{C} \sum_{c'=1}^{C} \mathbf{P}_{cc'} \cdot \ln \frac{\mathbf{P}_{cc'}}{\mathbf{P}_c \cdot \mathbf{P}_{c'}}.$

Image clustering and Image segmentation

- 分类训练：
  - 训练集没有标签→扰动策略：
  - 其中g是一个微小的变化

$$\max_{\Phi} I(\Phi(\mathbf{x}), \Phi(g\mathbf{x})),$$

- 分割训练：
  - 分割时基于patch进行的分类，每个patch的中心的类别是输出结果
  - 在g进行扰动的时候也会改变patch中心的位置
  - 每个patch进行g的扰动，等价于对全图扰动、卷积操作后取出对应为之的featuremap

$$\max_{\Phi} \frac{1}{|T|} \sum_{t \in T} I(\mathbf{P}_t), \tag{5}$$

$$\mathbf{P}_t = \frac{1}{n|G||\Omega|} \sum_{i=1}^{n} \sum_{g \in G} \overbrace{\sum_{u \in \Omega} \Phi_u(\mathbf{x}_i) \cdot [g^{-1}\Phi(g\mathbf{x}_i)]^{\top}_{u+t}}^{\text{Convolution}}.$$

Experiments-分类

	STL10	CIFAR10	CFR100-20	MNIST
Random network	13.5	13.1	5.93	26.1
K-means [53]†	19.2	22.9	13.0	57.2
Spectral clustering [49]	15.9	24.7	13.6	69.6
Triplets [46]‡	24.4	20.5	9.94	52.5
AE [5]‡	30.3	31.4	16.5	81.2
Sparse AE [40]‡	32.0	29.7	15.7	82.7
Denoising AE [48]‡	30.2	29.7	15.1	83.2
Variational Bayes AE [34]‡	28.2	29.1	15.2	83.2
SWWAE 2015 [54]‡	27.0	28.4	14.7	82.5
GAN 2015 [45]‡	29.8	31.5	15.1	82.8
JULE 2016 [52]	27.7	27.2	13.7	96.4
DEC 2016 [51]†	35.9	30.1	18.5	84.3
DAC 2017 [8]	47.0	52.2	23.8	97.8
DeepCluster 2018 [7]† ‡	33.4⋆	37.4⋆	18.9⋆	65.6 ⋆
ADC 2018 [24]	53.0	32.5	16.0⋆	99.2
IIC (lowest loss sub-head)	**59.6**	**61.7**	**25.7**	**99.2**
IIC (avg sub-head ± STD)	59.8 ± 0.844	57.6 ± 5.01	25.5 ± 0.462	98.4 ± 0.652

Table 1: **Unsupervised image clustering.** Legend: †Method based on k-means. ‡Method that does not directly learn a clustering function and requires further application of k-means to be used for image clustering. ⋆Results obtained using our experiments with authors' original code.

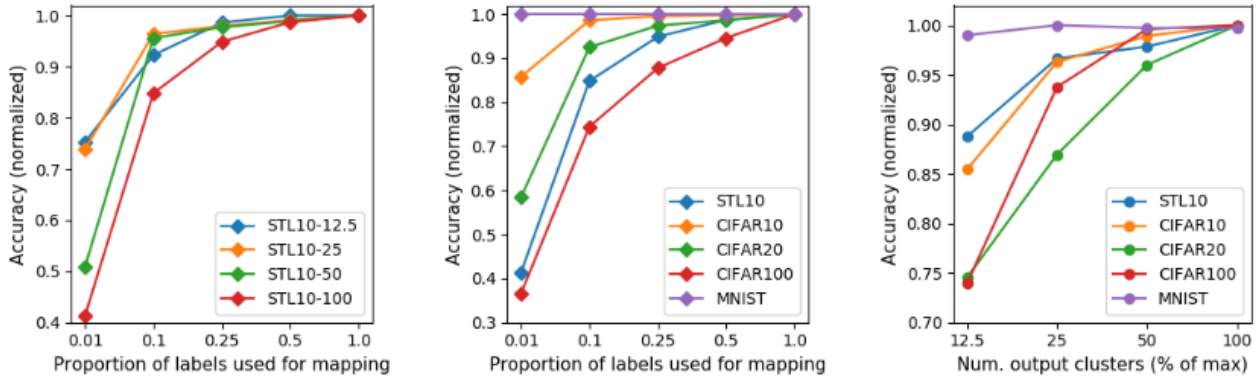
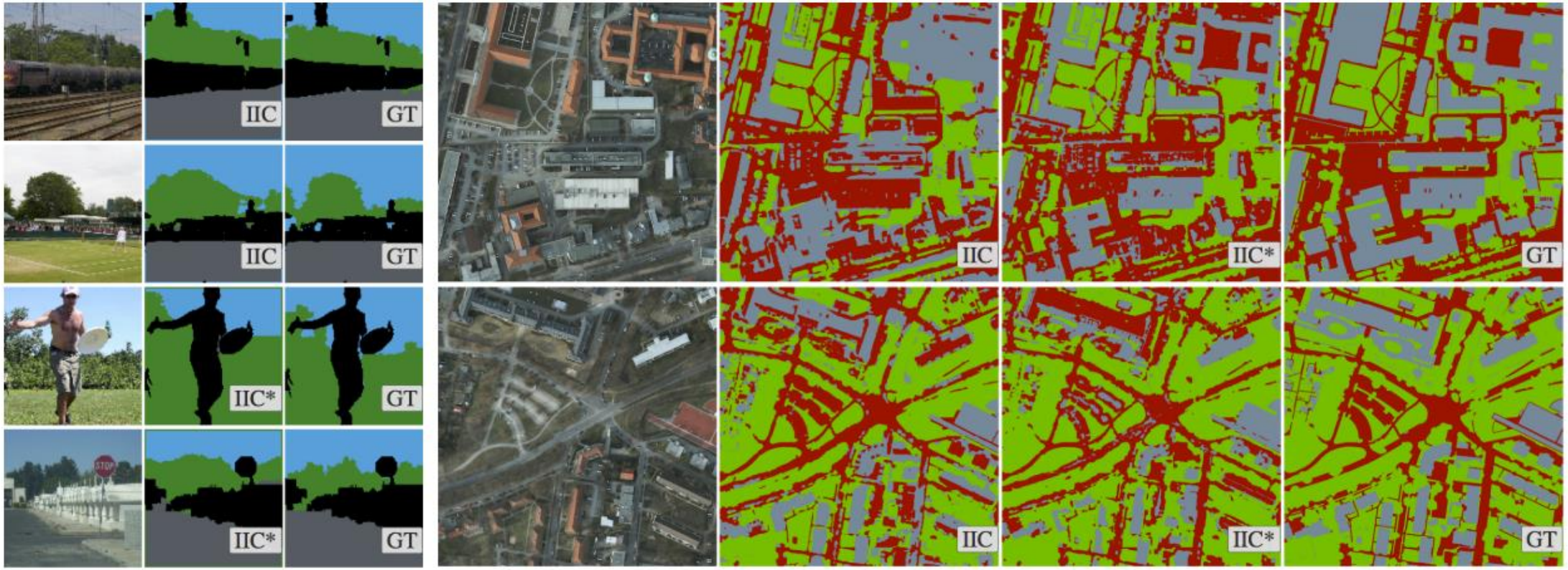


Figure 6: **Semi-supervised overclustering.** Training with IIC loss to overcluster ($k > k_{gt}$) and using labels for evaluation mapping only. Performance is robust even with 90%-75% of labels discarded (left and center). STL10-r denotes networks with output $k = \lceil 1.4r \rceil$. Overall accuracy improves with the number of output clusters k (right). For further details see supplementary material.

- 无监督：
  - 训练输出C类后，人为将C类和目标类别一一对应，得到正确率。
  - 采用多组初始化（sub-head）
- 半监督：
  - 半监督相当于利用部分标签提前构建出C类

Experiments- Segmentation



IIC (fully unsupervised segmentation) and IIC* (semi-supervised overclustering) results are shown, together with the ground truth segmentation (GT)

Experiments- Segmentation

	COCO-Stuff-3	COCO-Stuff	Potsdam-3	Potsdam
Random CNN	37.3	19.4	38.2	28.3
K-means [44]†	52.2	14.1	45.7	35.3
SIFT [39]‡	38.1	20.2	38.2	28.5
Doersch 2015 [17]‡	47.5	23.1	49.6	37.2
Isola 2016 [30]‡	54.0	24.3	63.9	44.9
DeepCluster 2018 [7]† ‡	41.6	19.9	41.7	29.2
IIC	**72.3**	**27.7**	**65.1**	**45.4**

Table 4: **Unsupervised segmentation.** IIC experiments use a single sub-head. Legend: †Method based on k-means. ‡Method that does not directly learn a clustering function and requires further application of k-means to be used for image clustering.