Semantic Segmentation Datasets for Resource Constrained Training

[Paper ID: 187]

Ashutosh Mishra[1*], Sudhir Kumar[1,2*], Tarun Kalluri[1,3*], Girish Varma[1], Anbumani Subramanian[4], Manmohan Chadrakar[3], CV Jawahar[1]

CVIT, IIIT Hyderabad[1]

University at Buffalo, State University of New York[2]

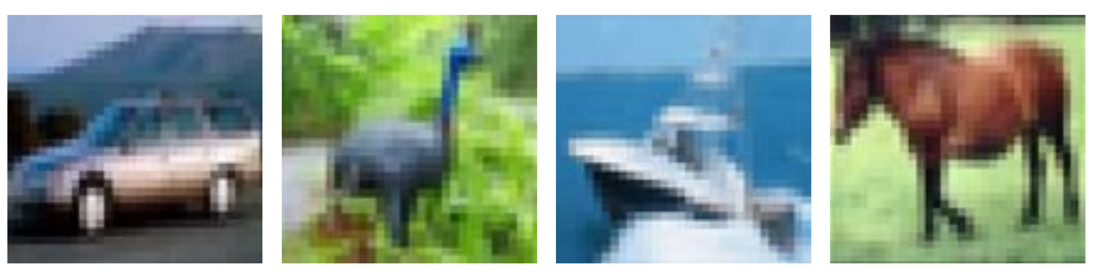
University of California, San Diego[3]

Intel Bangalore[4]

MOTIVATION

- Semantic Seg: Label each pixel !!
- Lot of datasets (IDD, BDD etc.)
- Current SS models:
  - Require higher GPU memory
  - Multi GPU Training !!

MNIST & CIFAR10(~174MB)

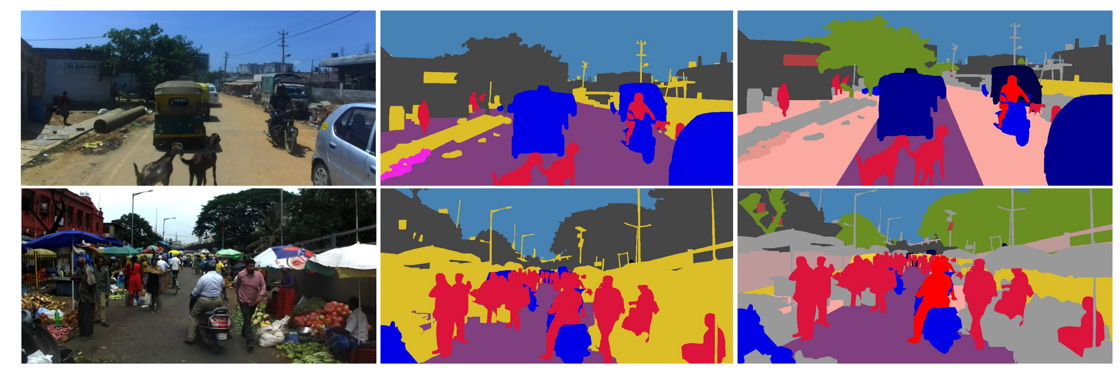


IDD & Cityscapes(**~30GB**)

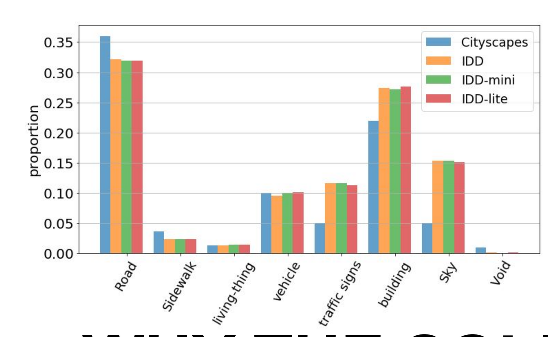
CHALLENGES

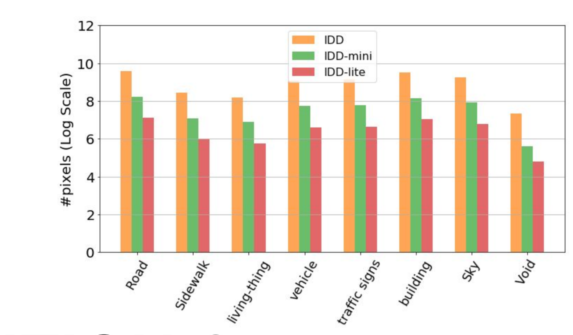
- Ubiquitous Adoption
  - Size and Scale
  - **30 GB vs 174 MB**
- Translation to bigger datasets
  - CIFAR10 := Imagenet
    - Liu *et al* 2017, Zhong *et al* 2018
  - Some works in SS
    - Chen *et al* 2018
  - But **problems in SS?**
    - Huge training cost !!
    - Resource intensiveness
  - **Solution**
    - Smaller datasets for quick prototyping
    - Replicate (eg IDD, Cityscapes)
- Limited labelled data
  - Why ?
  - Annotation time taking process
    - CS(~45 min), IDD(~1.15 hrs)
    - Learn using limited labelled data
  - **SOLUTION ?**

IDD MINI & IDD LITE



IDD Mini	IDD Lite
LRP	Teaching purpose
L2 labels	L1 labels
720(High) ASR	320 (High) ASR
IDD: #Img/4	IDD: #Img/10
1794 Tr 253 Val	673 Tr, 110 Val
~4 GB	< 1.5 GB





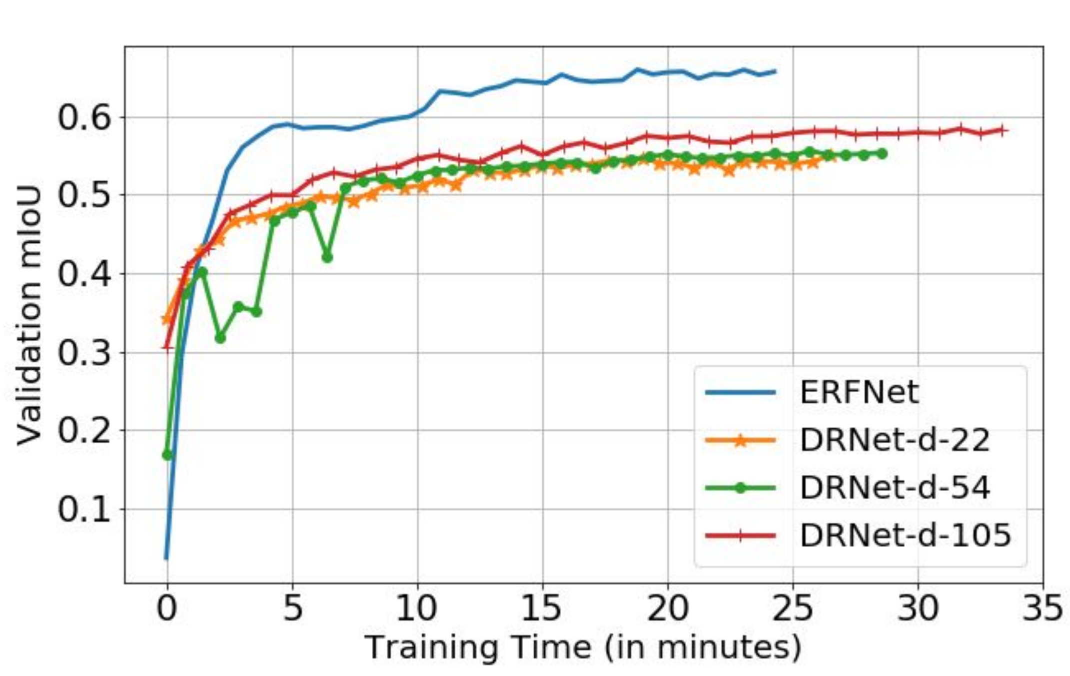
- **WHY THE SOLUTION ?**
  - RCS(Resource Constrained Settings)
  - Aimed for improving SS
    - Where ?
  - SOTA using limited labelled data
    - Less labels to full dataset (NAS)
    - Semi-supervised learning

EXPERIMENTS & RESULTS

Performance Benchmarking

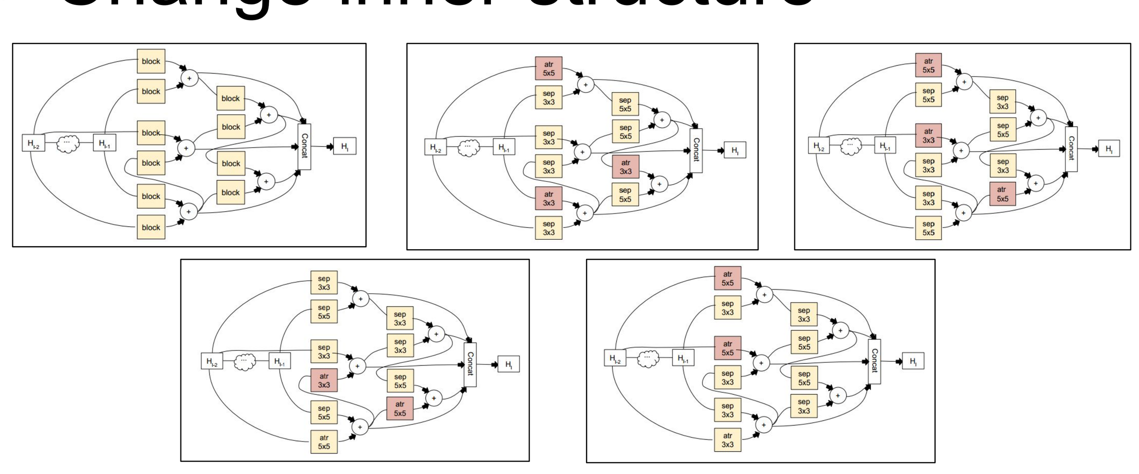
- ERFNet & DRNet(Resnet-18)
- Decent mIoU
  - 15 min training
  - 4 GB GPU
- Good for SOTA ? Teaching ?

Dataset	#L	Val. Res.	mIoU ERFNet	mIoU DRNet
CS	20	512x1024	71.50	68.00
IDD	26	512x1024	55.40	52.24
IDD Mini	16	480x640	57.91	53.31
IDD Lite	7	128x256	66.14	55.03



Architectural Search

- Identify optimal structure
- Keep outer structure intact
- Change inner structure



Layer	Type	Layer	Type
1	**Downsampler block**	1	**Downsampler block**
2	**Downsampler block**	2	**Downsampler block**
3-5	**3 x Conv-module**	3	**1 x Conv-module**
5-7	**2 x Conv-module**	4	**1 x Conv-module**
8	**Downsampler block**	5	**Downsampler block**
9-16	**8 x Conv-module**(dilated)	6	**Conv-module**(dilated 2)
17	**Deconvolution**(upsampling)	7	**Deconvolution**(upsampling)
18-19	**2 x Conv-module**	8	**1 x Conv-module**
20	**Deconvolution**(upsamling)	9	**Deconvolution**(upsamling)
21-22	**2 x Conv-module**	10	**1 x Conv-module**
23	**Deconvolution**(upsampling)	11	**Deconvolution**(upsampling)

Models	IoU(CS)	IoU(IDD-lite)
A*	64.54	58.15
B*	59.21	56.93
C*	55.96	55.46
D*	52.35	53.64

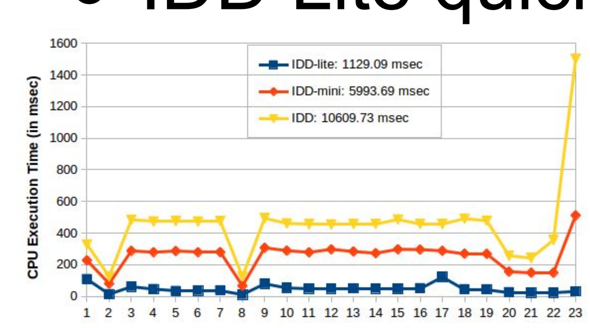
Models	IoU(CS)	Params	IoU(IDD-lite)
ERFNet	70.45	2038448	53.975
D*	68.55	547140	52.01
DG2*	65.35	395568	50.71
DG4*	61.42	319792	48.88
DG8*	59.15	281904	46.40

Correlation to Eff. SS models

- Efficient Seg Modules
  - Lower compute needs
  - Decent IoU
  - Howard *et al*, Zhang *et al*

Models for RPi

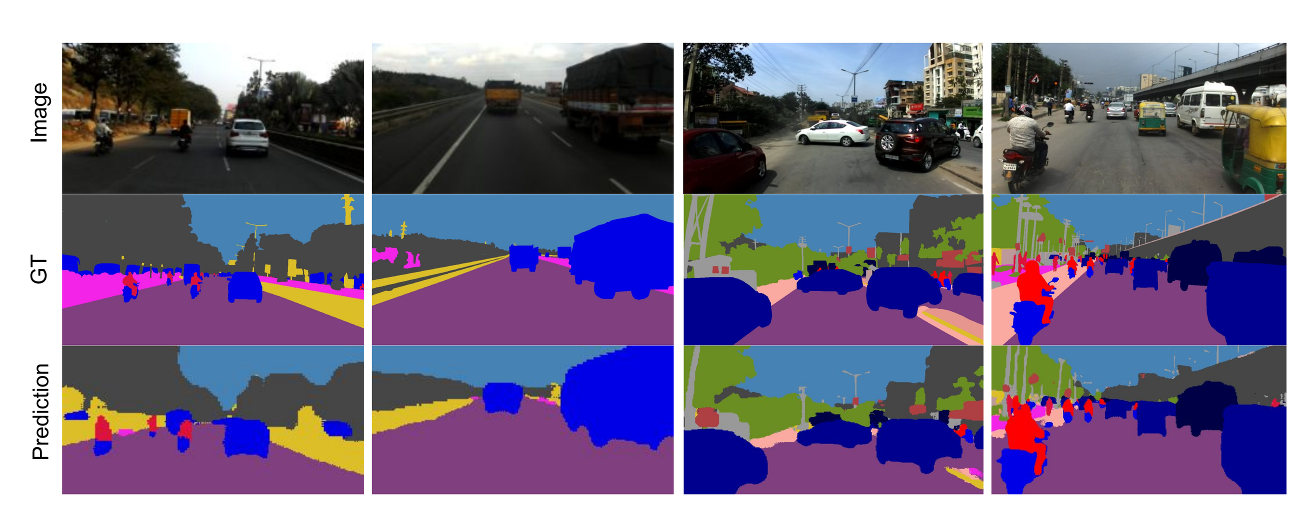
- Why Real Time deployment ?
  - Performance
  - Low power consumption
  - Feasible hardware cost
- Why RPi
  - Low cost, easily available
- IDD Lite quick for prototyping



	IDD-lite	IDD-mini	IDD
ERFNet	1.12	5.99	10.60
DRNet-18	56.61	78.47	95.94

- IDD Lite: time less for all layers & consistent across layers

IDD Mini **IDD Lite**



*** equal contribution**