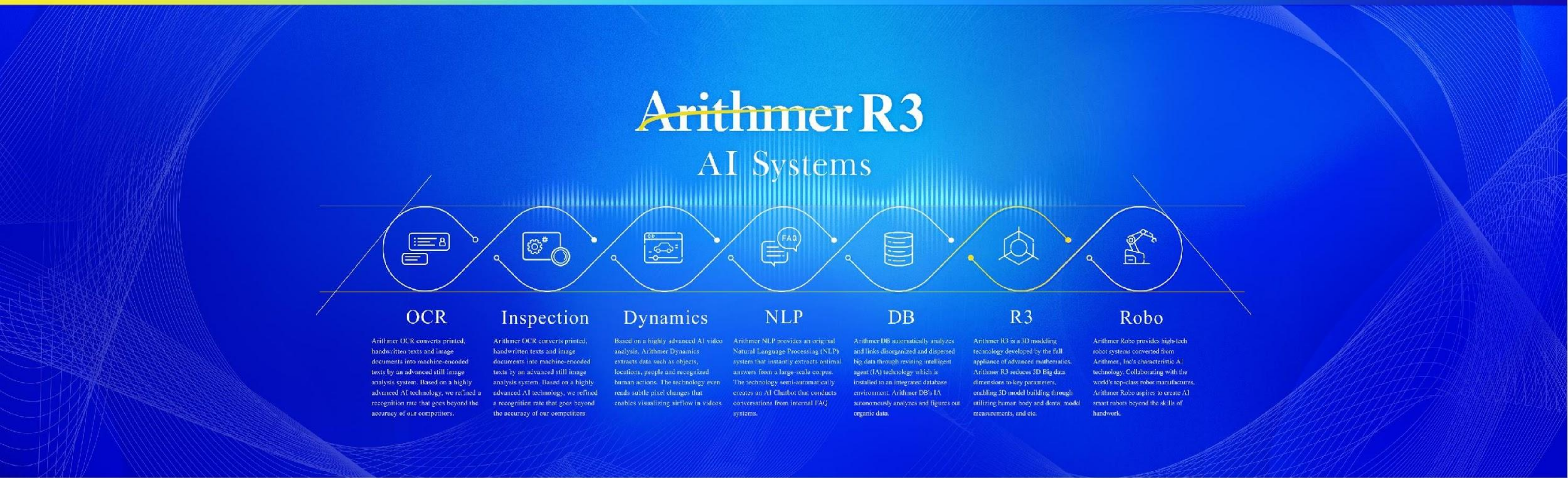


Working Group
Survey on Self-Supervised Image Segmentation

Arithmer R3 Div. R3 - Yann Le Guilly

2020/07/02

Self-Introduction: Yann Le Guilly

- Education
  - Undergraduate in physics, University of Rennes 1 (France)
  - Master’s degree in applied mathematics, IRMAR laboratory, University of Rennes 1 (France)
  - Research Student at Tokyo Institute of Technology, Murata Laboratory

- Former Job
  - Machine Learning Engineer at Abeja, Inc.
    - Development of AI-based product and Proofs of concept
- Current Job
  - Machine Learning Engineer
    - Development of AI-based product and Proofs of concept

Agenda

- Introduction
  - Image Segmentation
  - Problems
  - Self-supervised learning
  - Content of this presentation
- Survey
  - Pseudo Labeling
  - Class Activation Map
  - Image Depth Information
  - How about using videos
- Some comments

Introduction: Image Segmentation

Introduction: Problems

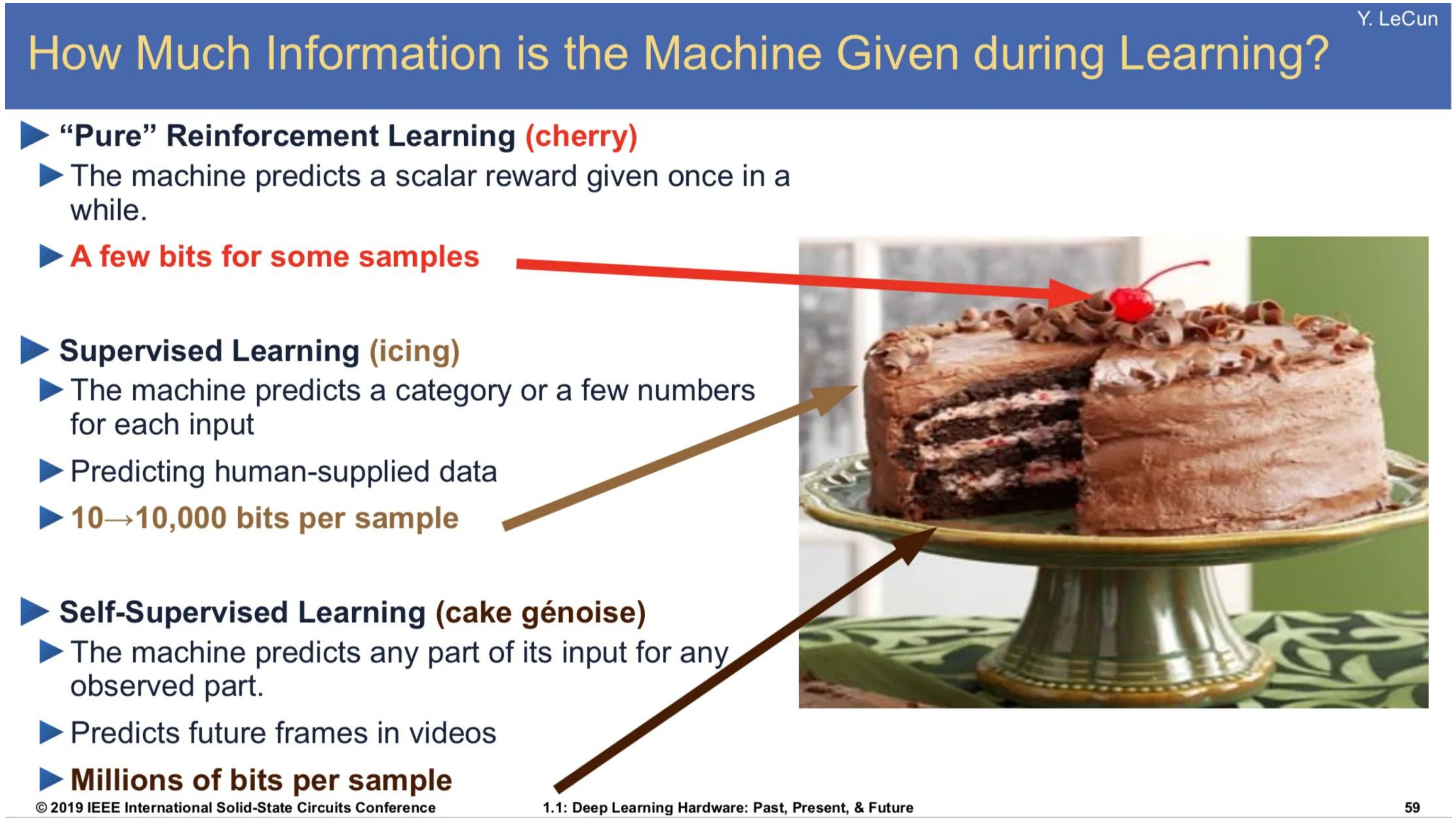
- very hard to annotate (=expensive)
  - pixel-wise annotation
  - different types of objects
  - boundaries are often blurry
- easy to make mistake
  - lots of unclear cases
  - Need to keep focus for long time

How can we change this situation?

CVAT: open source annotation tool

Introduction: Self-supervised learning

- Research topic pushed by "godfathers of AI" and specially by Yann Le Cun
- Pre-train a feature extractor in an unsupervised way then train the classifier with annotated data
- Reach SOTA with only 10% of labels in the last papers (starting to go beyond)
- Methods coming from NLP



Slide from Yann Le Cun's (recurrent) presentation

Introduction: Content of this presentation

- 3 very different directions
  - Using **pseudo labelling**
  - Using **Activation Map** (like grad-CAM)
  - Using **depth information**
  - Leverage frames in **videos**

Spoiler: there is no self-supervised learning semantic segmentation method that is very convincing...

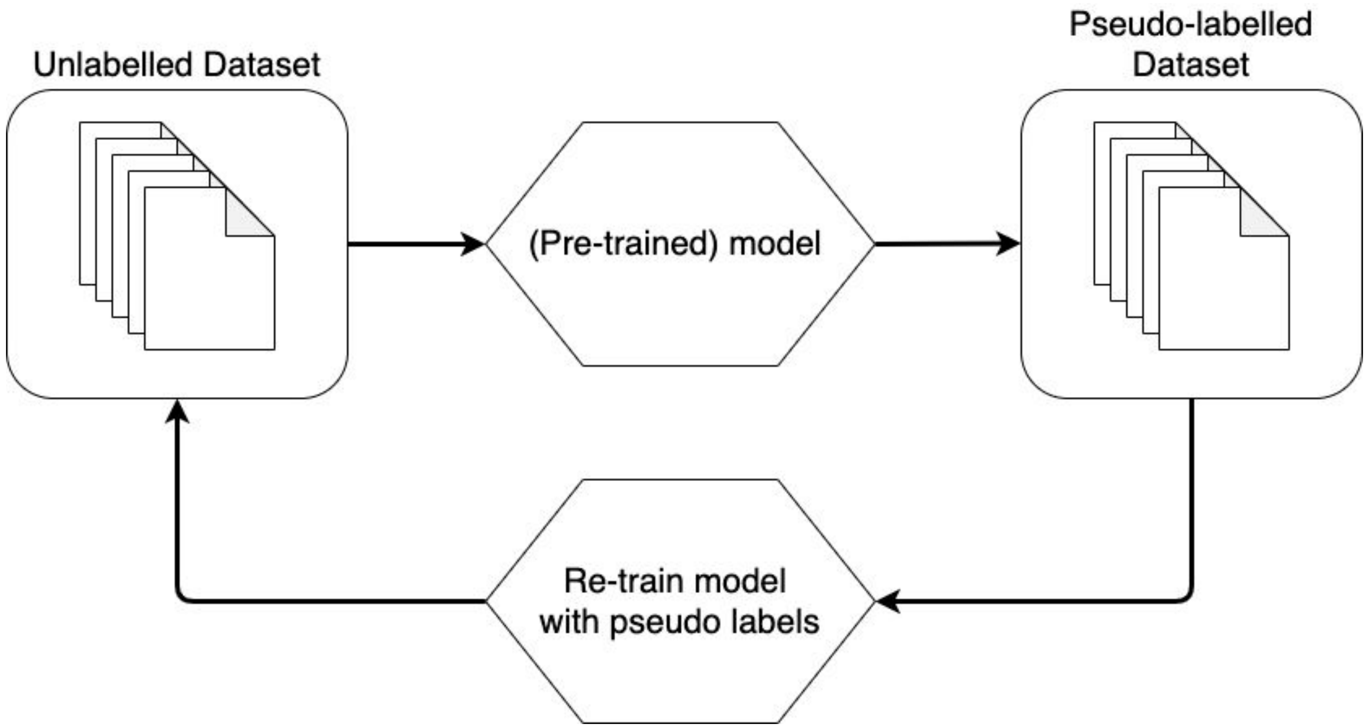
- High level explanations
- For more details, I provided notes for some papers
- For even more details, please read the original papers

Pseudo Labeling: Introduction

- Usually use softmax to differentiate “strong prediction” to “weak prediction”
- Keep only “strong prediction” as pseudo-labels
- popular research topic

Issue:

- Softmax is not the best way since it takes the best score without considering other scores (which might be promising also)



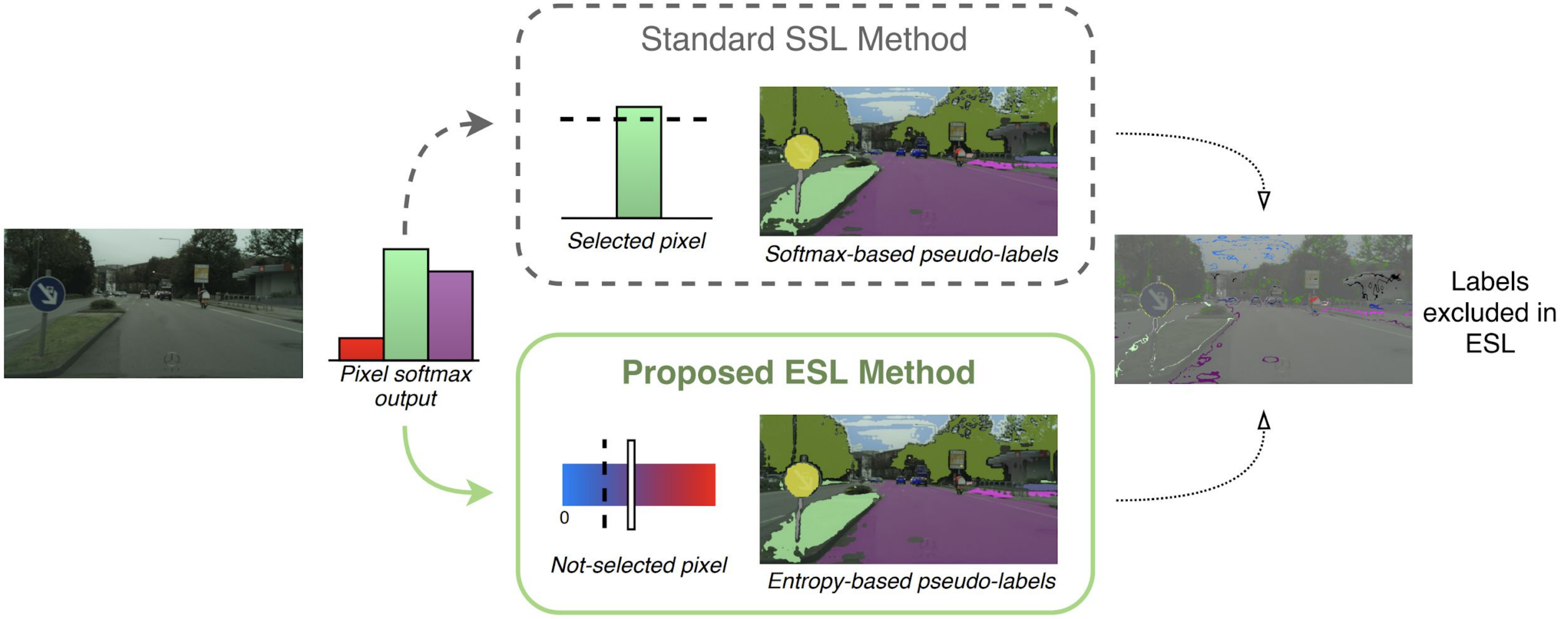
Simple illustration on what is pseudo labeling

Pseudo Labeling: Entropy-guided (1/2)

My notes:
https://arithmer.co.jp/wp-content/uploads/pdf/notes_ESL_Entropy-guided_Self-supervised_Learning_for_Domain_Adaptation_in_Semantic_Segmentation.pdf

Original paper: https://arxiv.org/abs/2006.08658v1

Repo: https://github.com/liyunsheng13/BDL



Pseudo Labeling: Entropy-guided (2/2)

GTA5 → Cityscapes

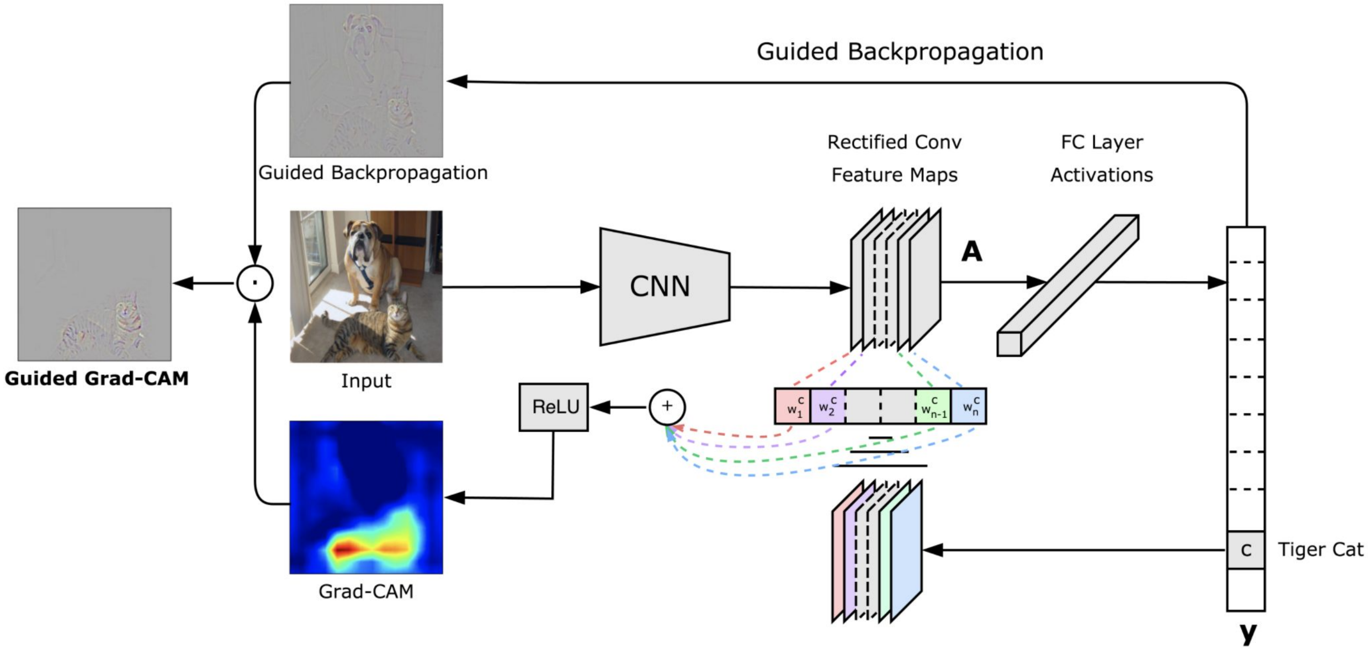
Method	Self-Training	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
AdaptSegNet [22]	-	79.7	16.4	76.1	18.8	12.7	24.8	33.3	20.8	**82.0**	**17.1**	73.4	55.8	27.3	62.3	**37.2**	30.0	**1.4**	30.8	15.1	37.6
	SSL	80.3	17.5	78.0	19.0	19.1	26.2	36.3	22.1	81.5	17.0	72.4	55.5	28.3	62.1	**37.2**	34.9	0.9	**31.4**	**20.2**	38.9
	ESL	**81.3**	**21.7**	**78.5**	**20.6**	**21.2**	**28.0**	**37.3**	**24.8**	81.1	16.1	**73.7**	**56.0**	**29.1**	**64.1**	34.0	**35.8**	0.9	**31.4**	19.2	**39.7**
ADVENT [23]	-	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	**34.1**	77.1	57.4	27.9	83.7	29.4	39.1	**1.5**	28.4	23.3	43.7
	SSL	89.6	35.4	82.0	**29.7**	25.6	31.9	36.6	25.6	**84.3**	29.7	75.2	59.9	**29.9**	84.7	**40.4**	42.6	0.1	**32.7**	30.9	45.6
	ESL	**90.0**	**38.6**	**82.9**	**29.7**	**28.3**	**33.2**	**38.5**	**25.8**	83.9	25.8	**78.3**	**60.0**	**29.9**	**85.9**	35.5	**43.3**	1.1	29.1	**32.0**	**45.9**
BDL [14] (step 1)	-	88.2	41.3	83.2	28.8	21.9	31.7	35.2	28.2	83.0	26.2	83.2	57.6	27.0	77.1	27.5	34.6	2.5	28.3	36.1	44.3
	SSL	87.3	39.4	83.7	30.2	24.9	**33.5**	41.0	30.8	83.3	27.9	83.4	58.7	29.9	75.2	28.4	33.3	1.9	30.1	33.0	45.0
	ESL	88.4	38.6	83.9	30.4	25.9	32.8	**41.5**	30.9	82.8	23.5	**85.3**	59.4	30.7	78.6	**30.8**	37.3	**4.5**	28.5	33.1	45.6
	SSL + IT	90.2	**47.5**	84.7	**33.6**	26.0	33.4	39.6	33.1	**84.1**	**33.8**	84.2	**59.9**	**31.5**	**79.8**	28.2	36.3	0.8	**31.5**	31.7	46.8
	ESL + IT	**90.3**	46.3	**84.8**	32.7	**26.9**	**33.5**	39.9	**34.8**	83.9	31.2	85.0	59.2	30.3	**79.8**	28.4	**43.4**	1.7	28.1	**36.2**	**47.2**
BDL [14] (step 2)	-	**91.0**	**44.8**	83.9	32.0	24.6	29.5	34.4	30.8	84.3	39.3	83.9	56.8	29.7	**83.3**	35.4	**49.8**	0.2	27.3	**37.1**	47.3
	SSL	89.7	39.6	84.1	30.2	28.4	**31.9**	39.0	29.4	83.9	35.1	**85.7**	58.0	31.6	80.8	36.2	46.6	0.5	28.9	33.7	47.0
	ESL	90.0	39.2	84.3	32.0	**31.1**	31.7	**39.2**	32.1	83.6	31.5	84.9	58.5	31.7	82.9	**39.5**	48.4	0.9	30.5	33.0	47.6
	SSL + IT	90.3	43.6	84.4	32.3	28.8	31.5	37.1	**34.2**	**84.7**	**42.3**	84.0	58.2	**32.3**	82.5	35.7	48.9	**1.9**	30.5	31.7	48.2
	ESL + IT	90.2	43.9	**84.7**	**35.9**	28.5	31.2	37.9	34.0	84.5	42.2	83.9	**59.0**	32.2	81.8	36.7	49.4	1.8	**30.6**	34.1	**48.6**

Model Pre-trained on GTA5 dataset and method used on Cityscapes: consistently improves the final accuracy but by less than 1% mIoU
State-of-the-art for fully supervised learning: 85.1% (IT = image translation)

Class Activation Map: Introduction

original paper: https://arxiv.org/abs/1610.02391

- Uses gradient information flow to the neurons of a specific CNN layer to identify regions of activation
- Step forward interpretability
- Work from 2017 (last version from last December)



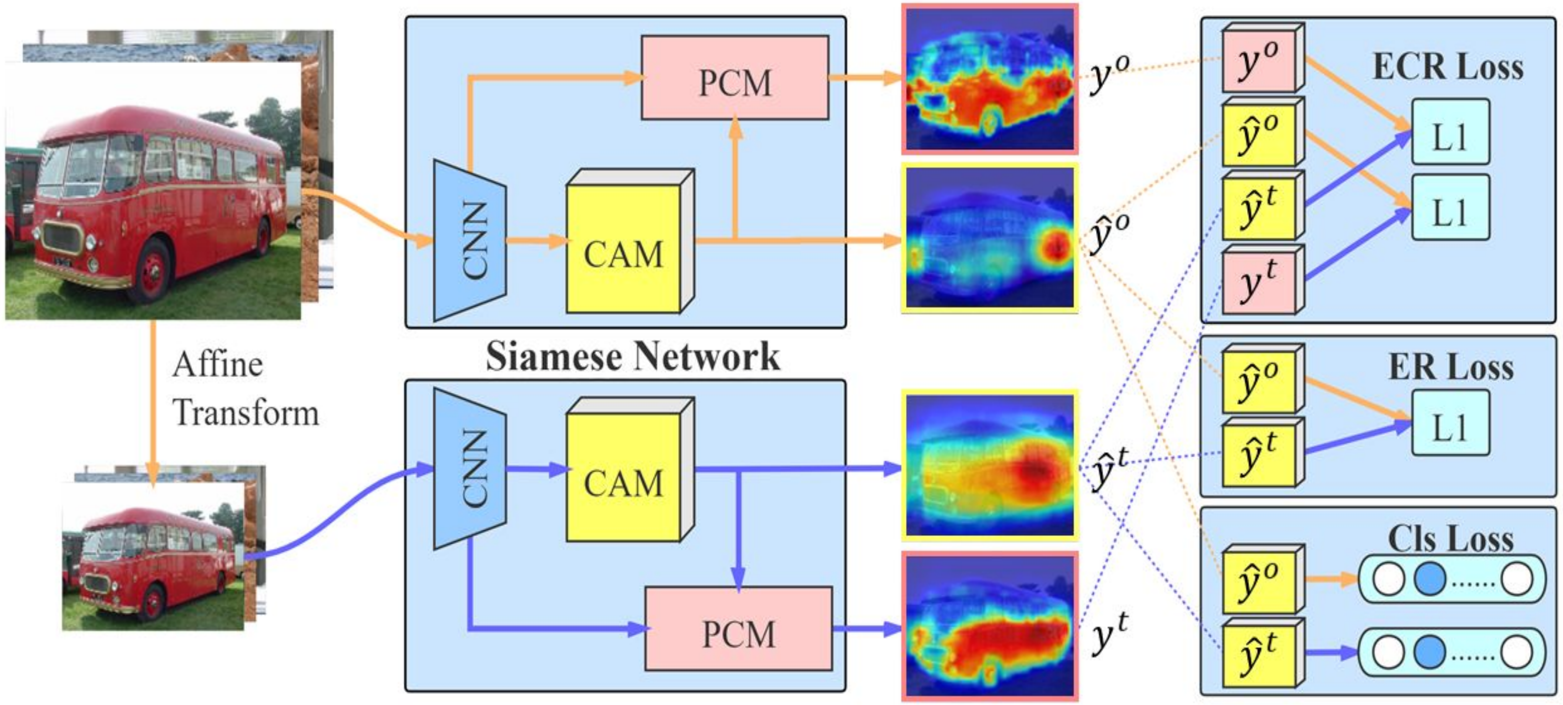
Class Activation Map: Equivariant Attention Mechanism

My notes: https://arithmer.co.jp/wp-content/uploads/pdf/notes_Self-supervised_Equivariant_Attention_Mechanism_for_Weakly_Supervised_Semantic_Segmentation.pdf

Original paper: https://arxiv.org/abs/2004.04581

Repo: https://github.com/YudeWang/SEAM

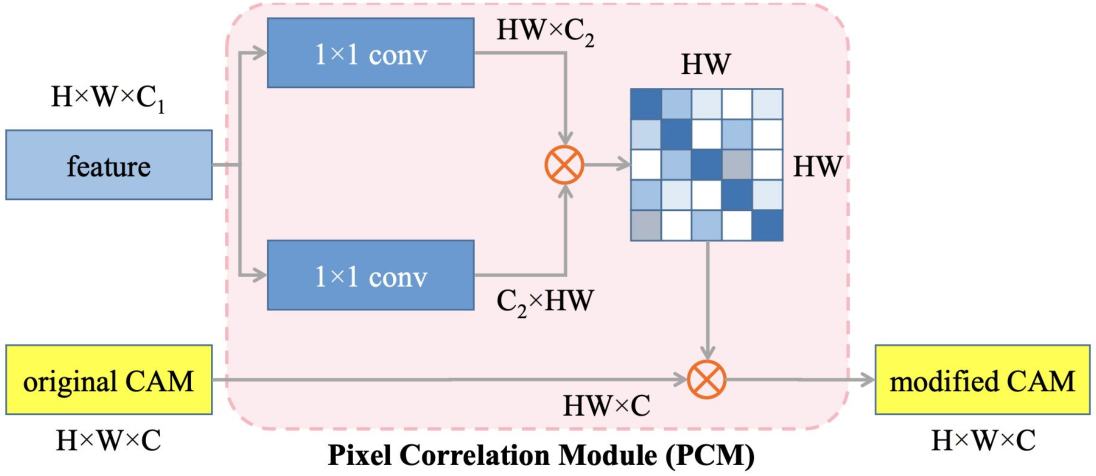
1. Need image level annotation (class)
2. 3 loss functions combines (ECR, ER and cls)
   a. both CAM outputs should be similar despite Affine transform
   b. But CAM degenerates (converge to a trivial solution) so ECR regularizes the PCM outputs with the original CAM



Class Activation Map: Equivariant Attention Mechanism

Main contribution: PCM module

- uses attention to refine the mask from grad-CAM
- attention can capture contextual information



Class Activation Map: Experiments

model	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIoU
CCNN [25]	68.5	25.5	18.0	25.4	20.2	36.3	46.8	47.1	48.0	15.8	37.9	21.0	44.5	34.5	46.2	40.7	30.4	36.3	22.2	38.8	36.9	35.3
MIL+seg [27]	79.6	50.2	21.6	40.9	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
SEC [19]	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
AdvErasing [32]	83.4	71.1	30.5	72.9	41.6	55.9	63.1	60.2	74.0	18.0	66.5	32.4	71.7	56.3	64.8	52.4	37.4	69.1	31.4	58.9	43.9	55.0
AffinityNet [2]	88.2	68.2	30.6	81.1	**49.6**	61.0	77.8	66.1	75.1	29.0	66.0	40.2	**80.4**	62.0	70.4	73.7	42.5	70.7	**42.6**	**68.1**	51.6	61.7
Our SEAM	**88.8**	**68.5**	**33.3**	**85.7**	40.4	**67.3**	**78.9**	**76.3**	**81.9**	**29.1**	**75.5**	**48.1**	79.9	**73.8**	**71.4**	**75.2**	**48.9**	**79.8**	40.9	58.2	**53.0**	**64.5**

Pascal VOC 2012 val with Image level annotation only (no mask). Using ResNet38 as backbone.

fully-supervised learning SOTA: 90.0%
(backbone is huge in this case though)

Comparison on Pascal VOC 2012 dataset with other weakly supervised methods

Methods	Backbone	Saliency	*val*	*test*
CCNN [25]	VGG16		35.3	35.6
EM-Adapt [24]	VGG16		38.2	39.6
MIL+seg [27]	OverFeat		42.0	43.2
SEC [19]	VGG16		50.7	51.1
STC [33]	VGG16	√	49.8	51.2
AdvErasing [32]	VGG16	√	55.0	55.7
MDC [34]	VGG16	√	60.4	60.8
MCOF [36]	ResNet101	√	60.3	61.2
DCSP [4]	ResNet101	√	60.8	61.9
SeeNet [15]	ResNet101	√	63.1	62.8
DSRG [16]	ResNet101	√	61.4	63.2
AffinityNet [2]	ResNet38		61.7	63.7
CIAN [10]	ResNet101	√	64.1	64.7
IRNet [1]	ResNet50		63.5	64.8
FickleNet [21]	ResNet101	√	64.9	65.3
Our baseline	ResNet38		59.7	61.9
Our SEAM	ResNet38		64.5	65.7

Image Depth Information: HN labels

- HN labels generation
    - Computing angles and height relative to the floor plane using RGB-D
    - Everything is binned to create labels
- Training segmentation model on HN-labels
- Fine-tuning on real dataset

The point is more to show:
pretrained HN labels >> pretrained ImageNet

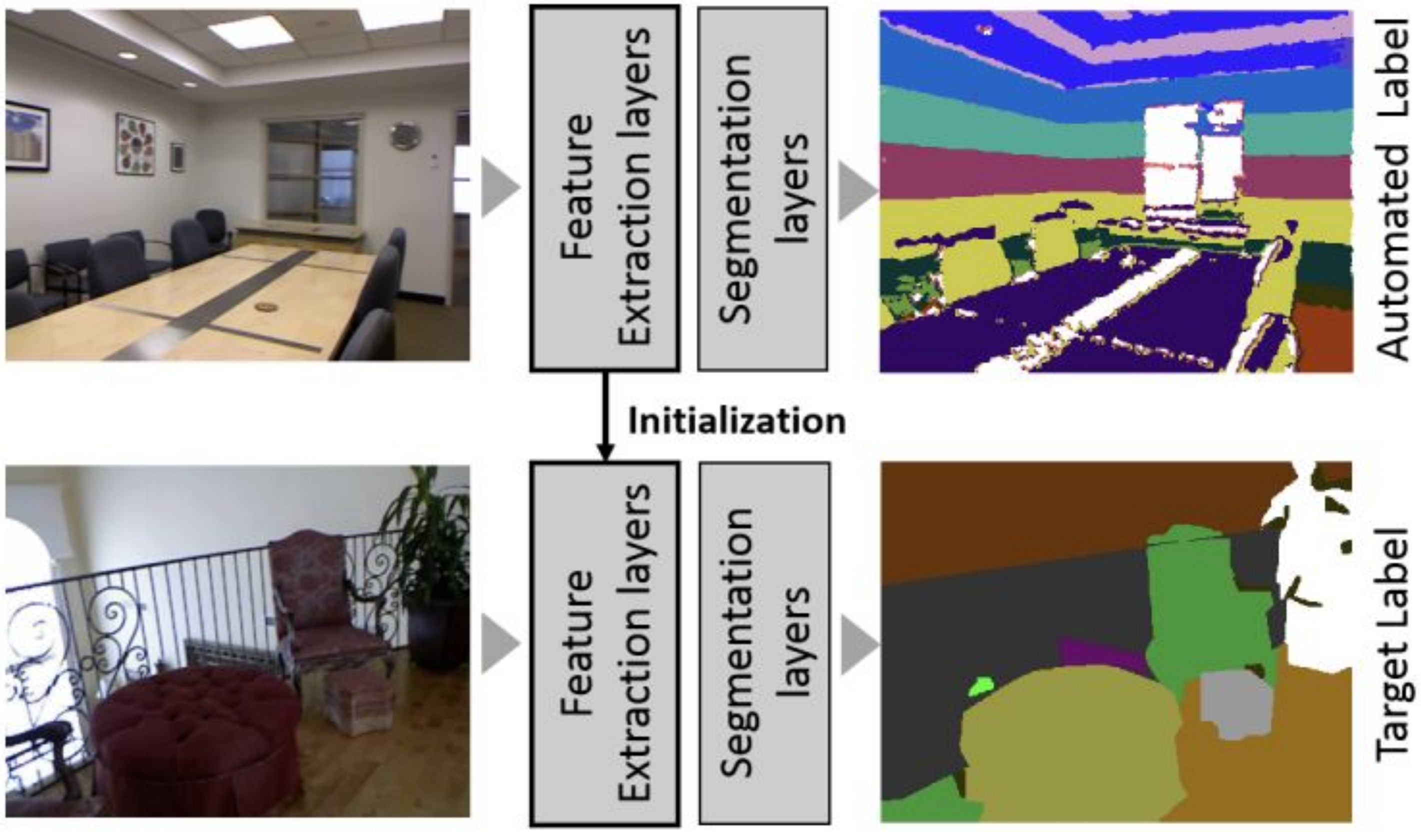


Image Depth Information: Experiments

Arch	Pre-training Method	Wall	Floor	Cabinet	Bed	Bookshelf	Sofa	Dresser	Avg acc	mIoU	Global acc
SegNet	No pre-training	77.57	74.91	42.36	13.25	6.58	22.42	3.17	13.73	8.78	38.74
	CIFAR100 pre-train	66.33	66.96	19.84	0.03	0.01	0.05	0	10.70	6.23	31.90
	ImageNet pre-train	81.54	86.1	**58.86**	37.76	35.05	**38.25**	4.35	25.40	17.34	50.58
	HN pre-training	**83.26**	**90.63**	58.81	**58.40**	**38.79**	37.05	**14.24**	**26.16**	**18.24**	**52.92**
DeepLab	No pre-training	18.70	66.72	60.78	22.30	20.70	15.24	12.49	24.77	6.73	31.41
	ImageNet pre-train	28.85	74.69	**87.98**	**64.40**	**56.22**	59.10	50.78	**57.84**	**34.27**	61.70
	HN pre-training	**32.00**	**77.64**	87.45	60.15	55.67	**60.51**	**53.69**	56.04	33.49	**62.98**

dataset: NUYv2, trained on 50 epochs only. HN labels are from NUYv2. ImageNet is 25x bigger

How about using videos: Self-supervised Video Object Segmentation

Original paper:
https://arxiv.org/abs/2006.12480v1

- Learn unsupervised to track similar pixels across frames
- Then when giving an initial mask (on this illustration, in the frame 0), the model can infer the same object for the next frames

How about using videos: Self-supervised Video Object Segmentation

Original paper:
https://arxiv.org/abs/2006.12480v1

- Learn unsupervised to track similar pixels across frames
- Then when giving an initial mask (on this illustration, in the frame 0), the model can infer the same object for the next frames

Method	Sup.	Overall	Seen		Unseen	
			$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Vid. Color. [3]	✗	38.9	43.1	38.6	36.6	37.4
CorrFlow [4]	✗	46.6	50.6	46.6	43.8	45.6
MAST [7]	✗	64.2	63.9	64.9	60.3	67.7
Ours	✗	**67.3**	**67.2**	**67.9**	**63.2**	**70.6**
OSMN [49]	✓	51.2	60.0	60.1	40.6	44.0
MSK [50]	✓	53.1	59.9	59.5	45.0	47.9
RGMP [51]	✓	53.8	59.5	-	45.2	-
OnAVOS [11]	✓	55.2	60.1	62.7	46.6	51.4
S2S [44]	✓	64.4	71.0	70.0	55.5	61.2
A-GAME [52]	✓	66.1	67.8	-	60.8	-
STM [40]	✓	79.4	79.7	84.2	72.8	80.9

On Youtube VOS val. J (Mean) J (Recall) F(Mean) F(Recall). The models in the bottom part of the table are trained fully-supervised.

Arithmer 株式会社
〒106-6040
東京都港区六本木一丁目６番１号 泉ガーデンタワー 38/40F(受付)
03-5579-6683
https://arithmer.co.jp/

Arithmer