Re-distributing Biased Pseudo Labels for Semi-supervised Semantic Segmentation: A Baseline Investigation

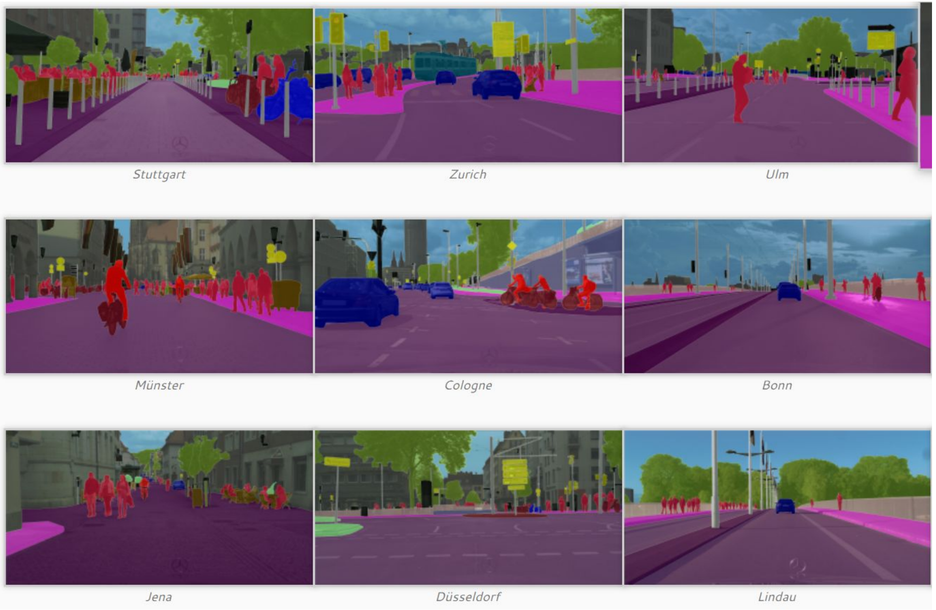
Ryhan Moghe

Problem

- Recently, DCNNs have been successful for semantic segmentation, but require lots of data with accurate pixel-by-pixel human annotations
- Self-training, by using semi-supervised learning with a small amount of labeled data to create pseudo-labels, then training on the pseudo-labeled data has achieved excellent results
- But most previous self-training approaches assume **class-balanced data distribution**, and use a single confidence scheme to create pseudo-labels, whereas most real-world datasets have long-tail class distributions (few categories make up majority of pixels)
- DCCNs trained on such a distribution will be biased towards the dominant categories

Data Sample Example

Cityscapes dataset samples



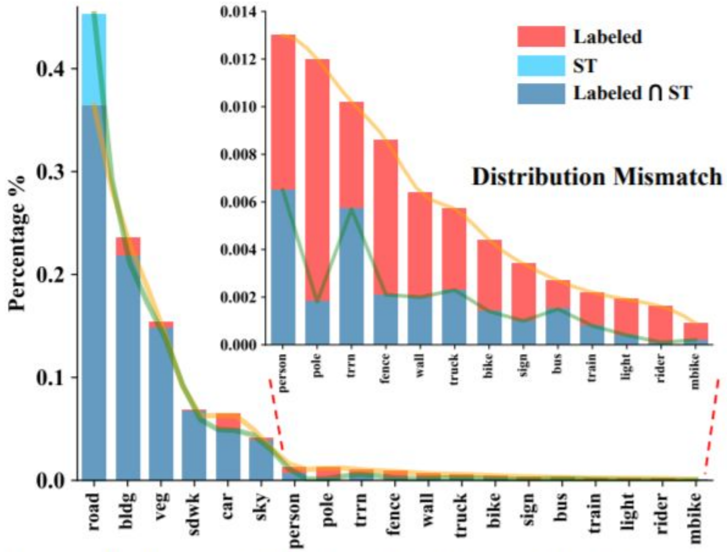


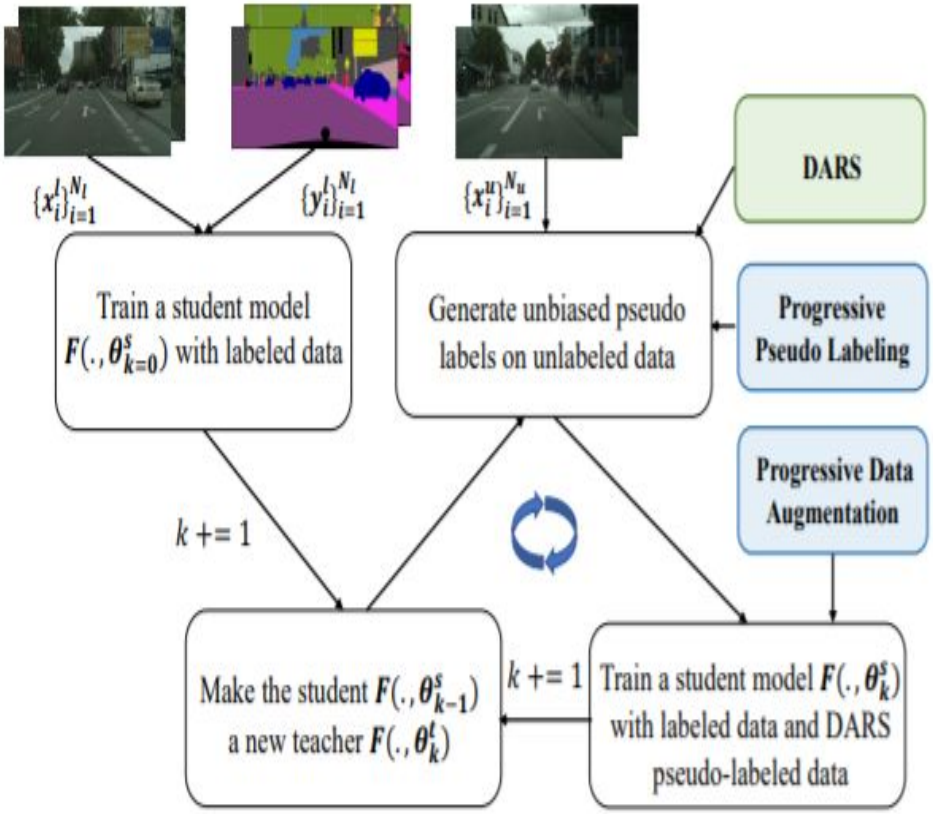
Figure 1. Class distribution mismatch on the Cityscapes dataset [10]. 'Labeled' and 'ST' denote the class distribution of true labels in the labeled set and pseudo labels produced by ST. We line up percentages of each class for better visualization.

Related Efforts

- Recent efforts attempt to address this issue by sampling the same % of pixels in each category based on predicted results, as opposed to a single confidence threshold
- But, the prediction class distribution will already deviate from true class distribution, thus pseudo labels will suffer the same bias
- Summary: Related works **do not exploit bias in pseudo-labeling**, and use ST for all classes or samples based on biased predictions, whereas this method **explicitly processes the bias in pseudo-labeling**

Method

- **Step 1**: Train student model F with labeled data, minimizing cross-entropy loss
- **Step 2**: Use student model F as teacher model to produce predictions for unlabeled samples
    - Using predictions and labels, generate pseudo labels using **DARS** method
- **Step 3**: Use labeled data and pseudo-labeled data to train F, minimizing cross-entropy loss for both labeled and unlabeled data
- Self-training iterates Steps 2 and 3 until no more performance gains



Unbiased Pseudo Label Generation

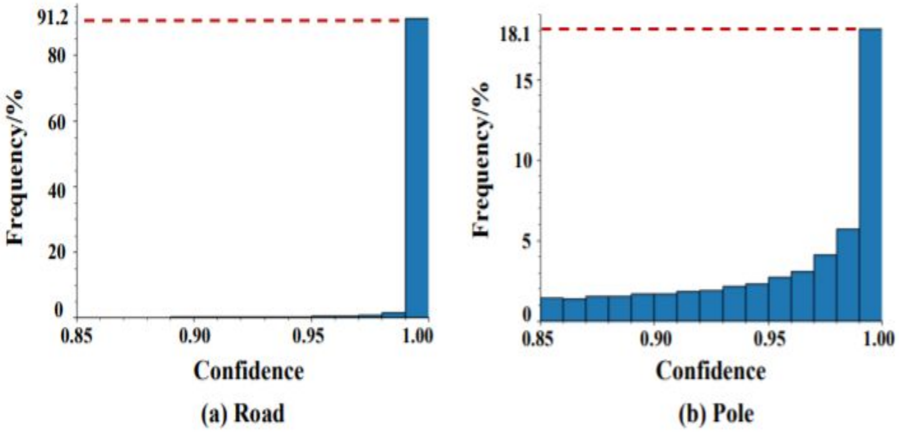
- Aim to obtain pseudo labels that occupy α% of all pixels (labeling ratio)
- Adopt category-specific confidence thresholds to derive pseudo-labels
- Confidence thresholds derived by finding thresholds that minimize KL-divergence of frequency of labels and pseudo-labels of each category:

$$\arg\min_{T} \; D_{\mathrm{KL}}(R(\{y_i^l\}_{i=1}^{N_l}), R(\{\tilde{y}_i^u\}_{i=1}^{N_u})),$$
$$\text{subject to } \; \tilde{y}_i^u = G(T, p_i^u), P(\{\tilde{y}_i^u\}_{i=1}^{N_u}) = \alpha\%.$$

R: Frequency function, outputs labeled/pseudo-labeled pixel % of each category
G: Generates pseudo-label for pixel if confidence value >= category threshold, otherwise assign ignore label
P: Returns percentage of pseudo-labeled pixels
*Pixels with ignore label will not contribute to training

Confidence Overlapping Issue

- In semantic segmentation, many pixels have similar/indistinguishable confidence values
- DCCNs are prone to producing over-confident prediction values, especially for head categories, causes these confidence overlaps
- This results in number of pixels after thresholding being larger/much larger than desired



Distribution Alignment and Random Sampling (DARS)

- Assume no confidence overlapping, and perform distribution alignment
- For categories that do not suffer serious confidence overlapping, we can derive the desirable number of pixels for each category j, by ignoring all pixels for category j with confidence lower than t_j

Algorithm 1 DARS

Input: Labeled set labels $\{y_i^l\}_{i=1}^{N_l}$, network predictions on the unlabeled set $\{p_i^u\}_{i=1}^{N_u}$ and labeling ratio α.

Output: Pseudo labels $\{\tilde{y}_i^u\}_{i=1}^{N_u}$

1: # Distribution Alignment

2: Calculate $\{n_i^u\}_{i=1}^{C}$, $\{t_j\}_{j=1}^{C}$ according to Eq. (3) and Eq. (4)

3: Obtain initial pseudo labels $\{\tilde{y}_i^u\}_{i=1}^{N_u}$ by ignoring low confidence labels in $\mathrm{argmax}\{p_i^u\}_{i=1}^{N_u}$ compared with $\{t_j\}_{j=1}^{C}$

4: # Random Sampling

5: Count sampling ratio: $\{s_j\}_{j=1}^{C} \leftarrow n_j^u / \tilde{n}_j^u$

6: Update $\{\tilde{y}_i^u\}_{i=1}^{N_u}$ by randomly ignoring $1 - s_j$ percent pseudo-labeled pixels for each class j

Progressive Data Augmentation and Labeling

- If we keep the labeling ratio and data augmentation magnitude the same, training loss starts low
- Increasing labeling ratio allows model to evaluate new data samples, but alone changes loss very little
- We further introduce new samples to our model by increasing magnitude of data augmentation

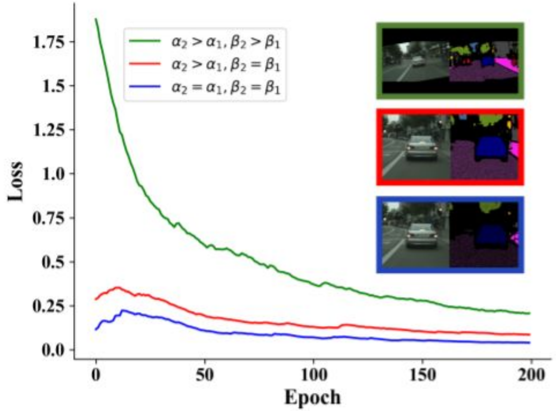


Figure 4. Training loss of pseudo labels in iterative training (k=2), where α_k and β_k denote the labeling ratio and the strength of data augmentation at round k. Right are examples of image crop and pseudo label pair for each case.

Experiment

- Dataset:
  - Cityscapes
    - 5k fine annotated images
    - 2975, 500, 1525 three image sets for training, validation, and testing
    - 19 urban-scene semantic classes defined for semantic segmentation
    - ⅛ and ¼ of training set are randomly sampled for labeled set, and remaining are used for unlabeled set
  - VOC12
    - 20 semantic classes, 1 background class
    - 1464 training set used as labeled data, 9k augmented set used as unlabeled data
- Architecture:
  - PSPNet:
    - Used instead of SOTA methods often contain heavy engineering and parameter tuning
    - Best trade off between reproducibility, performance, and costs

Comparison: DARS vs SOTA methods on Cityscapes

Method	Split	mIoU (%)			
		Baseline	Result	Oracle	Gain
Hung *et al.* [26]	1/8	55.5	58.8	67.7	3.3
	1/4	59.9	62.3		2.4
Mittal *et al.* [42]	1/8	56.2	59.3	65.8	3.1
	1/4	60.2	61.9		1.7
CutMix [19]	1/8	55.25±0.66	60.34±1.24	67.53±0.35	5.09
	1/4	60.57±1.13	63.87±0.71		3.30
DST-CBC [18]	1/8	56.7	60.5	66.9	3.8
	1/4	61.1	64.4		3.3
Mendel *et al.* [41]	1/8	55.96±0.86	60.26±0.84	66.9	4.3
	1/4	60.54±0.85	63.77±0.65		3.23
DARS (crop 361)	1/8	60.75±0.35	69.64±0.01	73.80±0.34	**8.89**
	1/4	66.54±0.48	71.30±0.08		4.76
DARS (crop 713)	1/8	65.54±0.34	**72.78±0.17**	76.60±0.67	7.24
	1/4	69.22±0.01	**74.32±0.12**		**5.10**

Table 1. Comparison with the state-of-the-arts on Cityscapes val set. DARS uses PSPNet50 backbone.

Method	mIoU
Deeplabv2 [6]	56.2
Hung *et al.* [26]	57.1
Mittal *et al.* [42]	59.3
CutMix [19]	60.34
DST-CBC [18]	60.5
Mendel *et al.* [41]	60.26
DARS	**64.20**

Table 2. Comparison with the state-of-the-arts with DeepLabv2 backbone in 1/8 split setting for Cityscapes.

Comparison: DARS vs SOTA methods on VOC12

Method	Backbone	mIoU
GANSeg ([52])	VGG16	64.10
AdvSemSeg ([26])	DeepLabv2-101	68.40
CCT ([45])	PSPNet50	69.40
DARS	PSPNet50	**73.89**

Table 3. Comparison with the state-of-the-arts on VOC12 val set.

Ablation Studies

<u>Methods Tested</u>:

- ST (baseline): single confidence thresholding method
- CBST (SOTA) : class balanced confidence thresholding method, based on prediction results
- DA: proposed distribution aligning method without random sampling
- TS: temperature scaling, incorporating with DA, CBST, or ST to facilitate distribution alignment by calibrating model predictions

Ablation Study Results

Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	Tail mIoU	mIoU	Gain
Baseline	96.7	75.4	88.2	35.2	35.0	45.8	50.3	63.0	89.6	53.9	92.3	72.1	46.7	90.0	38.4	47.9	33.0	33.7	67.0	47.2	60.75$\pm$0.35	0.0
ST	**97.4**	**78.3**	89.5	43.4	38.1	47.6	55.7	68.1	**90.9**	56.6	**93.3**	**75.1**	51.4	**91.7**	49.0	67.9	38.0	46.5	**69.9**	54.0	65.70$\pm$0.40	4.95
ST + TS	**97.3**	77.7	89.5	43.2	39.1	48.3	57.8	68.7	**90.8**	56.3	**93.4**	74.9	50.7	91.4	48.5	67.9	40.2	46.3	**69.8**	54.3	65.89$\pm$0.30	5.14
CBST	97.1	77.6	89.5	42.4	44.9	50.2	**58.9**	**69.8**	90.6	**57.1**	93.1	**75.2**	**52.6**	91.5	49.2	68.4	35.5	46.3	69.6	55.0	66.29$\pm$0.05	5.54
CBST + TS	96.8	77.1	89.4	43.7	44.2	50.1	58.6	68.9	90.4	55.5	92.7	75.0	**52.6**	91.3	48.8	66.9	42.7	**51.1**	69.7	55.5	66.61$\pm$0.20	5.86
DA + TS	97.2	**77.9**	**89.7**	**44.8**	**45.6**	**50.7**	**59.2**	69.1	90.6	56.1	93.0	**75.1**	**52.9**	91.6	**54.8**	**69.4**	**43.1**	48.3	69.6	56.7	67.31$\pm$0.12	6.56
DARS	97.1	77.7	**89.8**	**50.2**	**46.3**	**50.8**	58.6	**69.5**	90.7	**57.4**	92.8	75.0	**52.6**	**91.8**	**57.6**	**70.3**	**44.3**	**49.9**	69.7	**57.9**	**68.01$\pm$0.12**	**7.26**
ST + IT	97.5	78.8	89.6	43.4	38.5	47.2	55.1	69.4	90.9	56.1	93.3	75.1	51.4	91.9	49.5	67.5	47.3	52.3	**70.4**	55.2	66.59$\pm$0.14	5.84
CBST + IT	**97.8**	**79.2**	**90.4**	44.6	**48.1**	**51.7**	59.8	**70.3**	**91.1**	58.1	**93.5**	74.9	52.7	**92.6**	53.1	70.5	24.2	53.6	**70.4**	56.1	67.20$\pm$0.38	6.45
DARS + IT	97.2	78.5	90.1	**49.3**	47.7	50.9	**59.9**	70.1	90.8	**59.6**	92.9	**75.2**	**54.4**	92.5	**67.7**	**73.0**	**48.7**	**54.7**	69.9	**60.6**	**69.64$\pm$0.01**	**8.89**

Table 4. Ablation study for different pseudo-labeling methods. The upper part reports results in a single self-training round (k=1, labeling ratio α =20%), and the lower part reports results with iterative training (IT). The tail classes are highlighted in blue. We make the top-2 results bold for the upper part, and top-1 bold for the lower part. Tail mIoU shows the mean IoU of tail classes.