> Tweak LAMBDA\_MIX to a higher value, as rn going forward Urban gets better and better (50% even) while Rural goes down: 25 is good (at least with LR set to e-5 or sth like that)

> Try adding weights: threshold weight is better than no weights

> Try ema: very nice

> Try tweaking alpha of ema: 0.99 was good enough

> Problem: too much shift. Possible Solution: try to use NormalizeOnRural on all images to create more uniform mixes: shit

> Problem: pseudo-labels are too noisy. Solutions: use pixel weights AND soft assignment, which we’re already doing. To improve on it I want to try using gradual pixel weights instead of on/off wrt threshold: shit

> Problem: class imbalance in the target domain.

Possible Solution: try to take buildings (or even background) only from source (Urban), as on Rural they seldom appear, and other similar things: shit (both hard and soft)

* Extra important: try to do it from the other way around, so don’t take buildings from Rural, as they’re really all over the place: just slightly worse (‘Buildings’ better than ‘BuildingsAndRoads’)

Possible Sol2: weigh loss inversely to the frequency of classes: slightly worse

> fix problem for which class -1 may be chosen when creating mixmasks, this causes unclassifieds possibly being present in a spaghetti like manner instead of the usual big box: fixed

**+++ Problem: more than 1k out of the 1.3k images of Train/Rural have unclassified, which makes up almost 20% of the total pixels of that folder. Consider running a preprocessing to create at least unclassified masks, which is easy to do as it’s pretty much the only thing that is pitch black (rgb #000000), and then use them a priori to set predictions to -1 regardless of what the model outputted, so that the loss backprop a 0 value on those pixels**

(in the first two of the following I use ema with apha=0.99 as the authors’ repo does)

> Tried ema/non ema in combination with 1 and 0.1, with no pxlwtd (threshold)

> Tried ema/non ema, with pxlwtd and with 0.1

> Now running alphas=[0.9, 0.999] and LRs[e-3, e-4], with ema pxlwtd and LAMBDA\_MIXED=0.1

NOTES:

> models that do not have it specified have the following hyperparams/settings:

* Don’t use the ema
* Use learning rate to e-4
* Use best\_model\_Resize as base (the old one, which actually had Resize+RC on non-virtual expansion)
* Pxlwtd uses hard threshold instead of soft weight based on confidence