III. Model documentation and write-up

You can respond to these questions in an e-mail or as an attached file (any common document format is acceptable such as plain text, DOCX, etc.) **Please number your responses.**

1. Who are you (mini-bio) and what do you do professionally?

   If you are on a team, please complete this block for each member of the team.

   I'm software engineer.

2. What motivated you to compete in this challenge?

   I'm using competitions mainly to test interesting papers (not everything is working out of box …), improve workflows.
   The competition has a nice goal, so a perfect playground.

3. High level summary of your approach: what did you do and why?

   Strait forward approach (and these days standard): end-to-end segmentation with fully convolutional neural network:

step 1:

- Net architecture: FPN with efficient-net(b1) back-bone (choice because of my machine for experiments is old-school box with GeForce GTX 1060)
- Identify hyper-parameters (weight decay, learning rate)
- train (one-cycle learning policy, optimizers: AdamW/Over9000, losses: Focal Loss, Dice Loss (in separate heads = train only once, see what work best, try to use all outputs in final ensemble))
- the network have additional 2 heads – direct classification head (is there a building or not – later to be used for negative examples mining) and scale regression head (the samples in dataset are in different scales, so this was a attempt to make things easier/better performing in inference)
- datasets: tier-1, and some data from SpaceNet dataset (5 fold)

step 2: Try “Self-training with Noisy Student”, and negative mining

- label test data and data from tier2 (soft labels) with model from step 1 (with TTA)
- mine negative samples
- repeat training with efficient-net(b2) back-bone, for soft labels, the KL divergence loss has been used.

Notes: The results on validation set and public leader-board improved against the base-line (cca 1.5 on Jaccard index), but there is no clear evidence that it is not just by using “stronger backbone” (classical mistake – try too many things in one step). I would not use these steps in other than “competition” mode (specially the negative mining can be the problematic point - I'll share with the code some (perhaps useful) additional charts from the validation phase).

step 3: Standard “competition madness”:

- average ensemble of models (from steps 1,2) with TTA augumentation (scale, flipping)

4. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

   I don’t think there is any :(- except of the 2 best performing solutions, the performance of top 10 participants is almost same (= reasonably good network with the same data).

5. What are some other things you tried that didn’t necessarily make it into the final workflow (quick overview)?

   Scale regression: I was trying to regress the resolution of the image and use this information during test-time / ensembling – did not work – probably because of the heavy augmentation during the training.

6. Did you use any tools for data preparation or exploratory data analysis that aren’t listed in your code submission?

   No.

7. How did you evaluate performance of the model other than the provided metric, if at all?

   By Focal Loss value.

8. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?

   No – just see the notes in section “Q2, step 2”.

9. Do you have any useful charts, graphs, or visualizations from the process?

   Not really – just std. learning curves.

10. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?

    ML is advancing fast – I’m looking forward to see some new techniques from transfer learning allowing to fit the data with less computational power involved.
    About the features/fields: I’m not sure – for competition, the data are ok (big scale / visual diversity), but on the other hand it makes problem unnecessarily harder – the data can normalized to common scale providing good context (when you see just the roof material without any context, no human can tell …) - so maybe the data from SAR sensor can make the problem solvable.

11. Did you learn new ways your machine learning skills could be applied to development (broadly defined by the Sustainable Development Goals) and/or disaster resilience and risk management more specifically? If so, what ways?

    I'm glad that a machine learning is used in such projects.