# 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.**

Currently I'm doing research at the xxx in Computer Graphics, using Machine Learning for problem solving, namely animate 3D character into depicted pose from sketch. I got my Master in Mathematics at the xxx, Russia. Lately, I fall in love with machine learning, so I was enrolled in xxx created by xxx company. This led me to develop and teach the first open deep learning course in Russian. Machine Learning is my passion and I often take part in the competitions. I think, Deep Learning eats the world.

2. What motivated you to compete in this challenge?

If I can share my knowledge with someone, I do it with pleasure. Secondly, I've never worked with geospatial data and technologies before, that's why this competition is a great opportunity to check my own skills and at the same time to get new knowledge from this field. You can also win some prize as a bonus:)

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

It turned out the approach is quite straightforward. You can get good results by training Unet-like models with heavy encoders using only tier 1 data and some tricks. The main idea is to take into account the rare tiles. One way is to assign some class to the tile and use the inverse probability of that class in the dataset to oversample them. It significantly speeds up the learning process. The other trick is to change binary cross-entropy loss to multiclass cross-entropy (in our case 2 output channels) and take argmax instead of searching an optimal threshold. It is also preferred to train the model in the same conditions as at inference, i. e. at inference we have 1024x1024 tiles, so we need maximally to preserve this resolution during training. To overfit on testset one you can do pseudo-labelling. After obtaining a strong single model do train 5 more models and do ensemble them by simple averaging.

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

I assign each tile a class `area_scene` and oversample rare classes which speed up learning significantly.
./src/train.py line 465
```
    proba = train_gps.y.value_counts().values
    proba = proba / proba.sum()
    to_weights = dict(zip(train_gps.y.value_counts().index, 1 / proba))
    train_gps['w'] = train_gps.y.apply(lambda x: to_weights[x])
```

```
    weights = torch.from_numpy(train_gps.w.values.astype('float32'))

    train_sampler = torch.utils.data.WeightedRandomSampler(weights, len(weights))
```

At the beginning I predicted 1-channel mask with binary cross-entropy loss and used default threshold 0.5. But with this configuration first 20-30 epochs jaccard index was around 0. Actually, optimal threshold was around 0.6. So, not to search this threshold I predict 2-channel mask with cross-entropy loss.
./src/train.py line 380

```
def BCEBCE(logits, target):
    return nn.CrossEntropyLoss()(logits, target[:, 0].long())
```

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

I tried to pretrain on tier 2 dataset, but the labels are not so good (shifted/not all building have labels), so it doesn't give me some advantages. Instead of 1-channel footprints I used 3-channel mask footprint/boundary/contact, but I didn't managed to get better results. Also I used knowledge-distillation and managed to train a single light model with 0.85 score.

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

No, I only see on the predictions of models and the distributions of jaccard index over tier 1 and tier 2. See here.

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

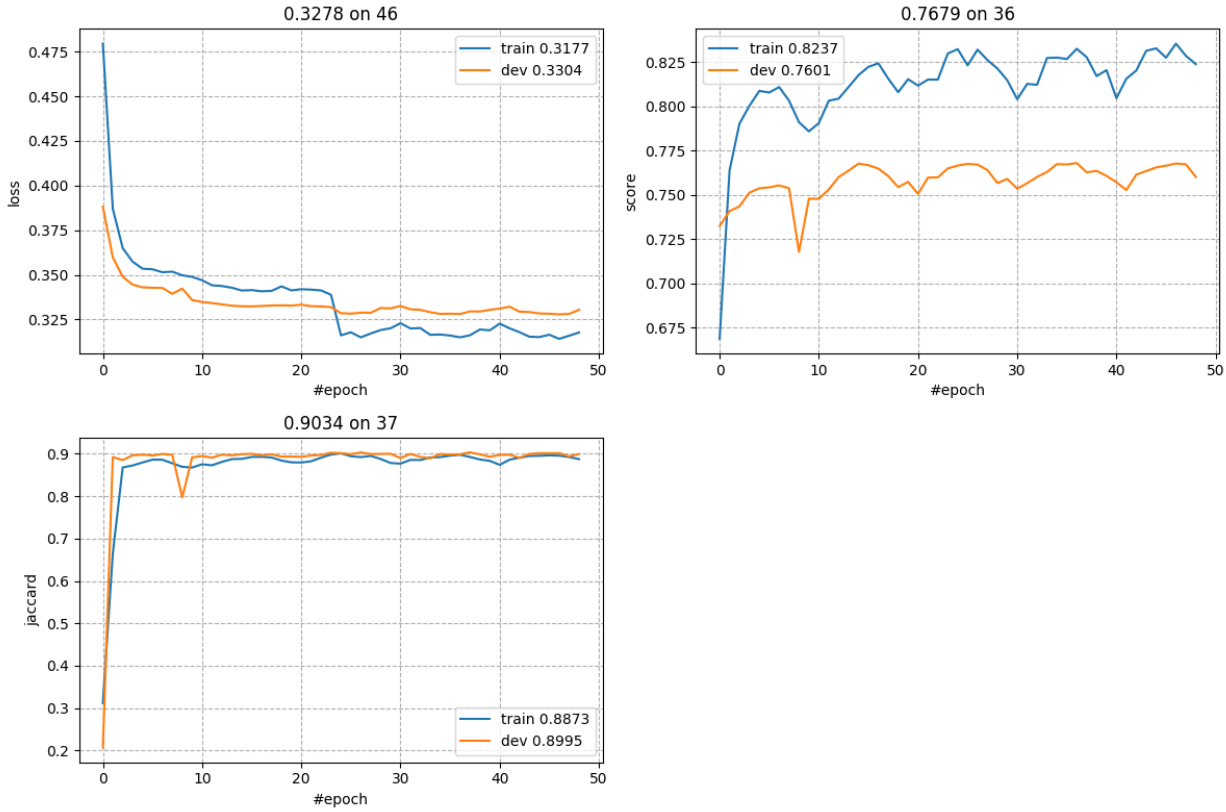
Only used the provided metric jaccard index.

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.)?

I trained models on 6 GPUs NVIDIA Tesla V100 with 32Gb RAM

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

All learning curves seems almost the same. Here is one of them:




`Score` is actually jaccard index micro, and `jaccard` is jaccard index macro.

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?

First, train using heavier CNN models like senet, se_resnext101 etc.
Second, clean or fix the labels from tier 2 dataset using strong models from tier 1. More ideas are here.

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?

See here.