NOVA course on deep learning in remote sensing

Home exercise

Raul de Paula Pires

# Objective

In this assignment, we aimed at comparing the effects of different annotation datasets for training a seedling detector. In addition, we explored different model sizes and hyper parameter configurations.

# Data

## Train/val data

* + **Own Annotations** (/content/drive/MyDrive/NOVA\_course\_deep\_learning/data/annotated\_data/train/ANNOTATOR\_ID) data prevenient from a single annotator on a limited amount of tiles from the training orthomosaic. 29 images were annotated and split in train (20 images) and validation (9 images).
  + **Full annotations** (annotations from all students merged). /content/drive/MyDrive/NOVA\_course\_deep\_learning/data/annotated\_data/train/full\_data. The 387 images were annotated and split in train (270 images) and validation (117 images).

## Test data

* + **Tiled test data** for evaluation using ML metrics. These data are located in /content/drive/MyDrive/NOVA\_course\_deep\_learning/data/annotated\_data/test
  + **Drone RGB orthomosaics** for evaluation using domain metrics. The files are stored in /content/drive/MyDrive/NOVA\_course\_deep\_learning/data/orthomosaics/test\_data.
    - In this exercise we evaluated the performance of the best trained detection in 3 sites: Galbyvien, Braatan and Holo. The site of Krakstad could not be used for prediction. In these sites, the tree positions were measured with GPS in the field (reference trees are in /content/drive/MyDrive/NOVA\_course\_deep\_learning/data/map\_data/test\_annotations2\_sun.geojson). The field data were collected for four square plots (approx. 0.1 ha) per site.

sites

Plots

# Methods

## Stats

Table 1 – Number of images and trees in each dataset

|  |  |  |
| --- | --- | --- |
|  | **Own** | **Full** |
| **n images** | 29 | 387 |
| **n trees** | 188 | 5074 |

## Model Training

In the in both own annotation and full datasets, we tested 3 different model sizes (YOLOn, YOLOm, YOLOx) with 2 different batch sizes (32 and 64) each, yielding a total of 6 combinations. Table 2 shows the mAP@.5 for all models.

With own annotations, the model with the largest [mAP@.5](mailto:mAP@.5) was YOLOm with batch size of 64, which had [mAP@.5](mailto:mAP@.5) equal to 0.5. For this reason, this model was chosen as the best model trained with the Own Annotations.

With the full annotations, the model with the largest [mAP@.5](mailto:mAP@.5) was YOLOm with batch size of 32, which had [mAP@.5](mailto:mAP@.5) equal to 0.3. For this reason, this model was chosen as the best model trained with the Own Annotations. There was not enough memory to train YOLOm with batch size 64 and YOLOx when using the full annotations.

Table 2 – [mAP@.5](mailto:mAP@.5) of the models trained with the own and full annotations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Own Annotations | | Full Annotations | |
|  | Batch size 32 | Batch size 64 | Batch size 32 | Batch size 64 |
| YOLOn | 0.26 | 0.26 | 0.26 | 0.26 |
| YOLOm | 0.505 | 0.505 | 0.30 | x |
| YOLOx | 0.193 | 0.193 | x | x |

## Model Evaluation – ML Metrics

The model trained with own annotations performed better when observing machine learning indicators, such as [mAP@.5](mailto:mAP@.5) (Table 2), confusion matrixes (Table 3) and F1-confidence curve (Figure 1). However, it is important to bear in mind a basic difference between the two datasets, the size. The full dataset is more than 10 times bigger than the own annotation dataset both in number of images and annotated seedlings. Thus, the higher accuracy in ML metrics in the observed in the model trained with own annotations could be an indication of overfitting.

Table 3 – Confusion matrixes for seedling detectors trained using the own and full annotation training datasets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Own Annotation | | Full Annotation | |
|  | Tree | Background | Tree | Background |
| Tree | 0.62 | 1 | 0.45 | 1 |
| Background | 0.38 | 0 | 0.55 | 0 |

|  |  |
| --- | --- |
| F1 curve Own Annotations | F1 curve full annotations |
|  |  |

Figure 1 – F1-confidence curve of the seedling detectors trained using the different training datasets.

## Model Evaluation – Domain Metrics

The predictions based on own annotations had bigger RMSE and bias than the predictions made using full annotations for training. Nevertheless, the predictions made using both models had high bias and RMSE values (table 4). In addition, both models systematically underestimate seedling density (Figure 2).

Table 4 – RMSE and bias of the seedling detector trained using the different training datasets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Own Annotation | | Full Annotation | |
|  | RMSE | bias | RMSE | Bias |
| Tree | 1030 trees/ha (82%) | -929 trees/ha  (-74%) | 509 trees/ha (40%) | -426 trees/ha (-34%) |

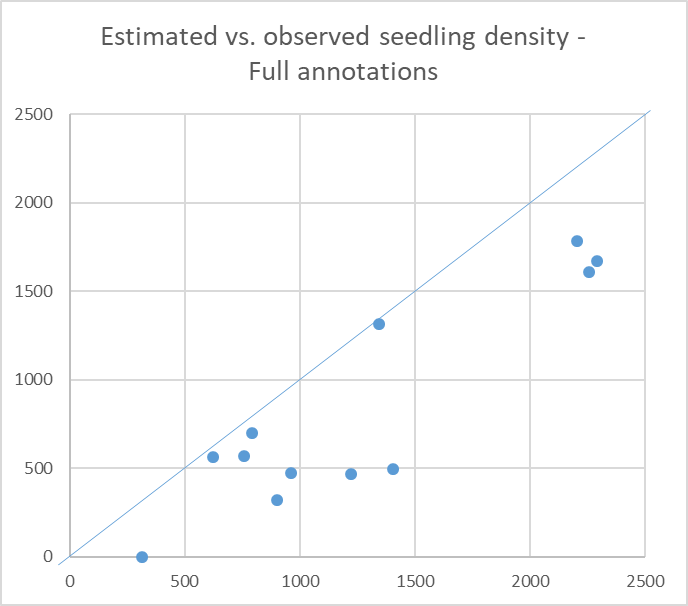
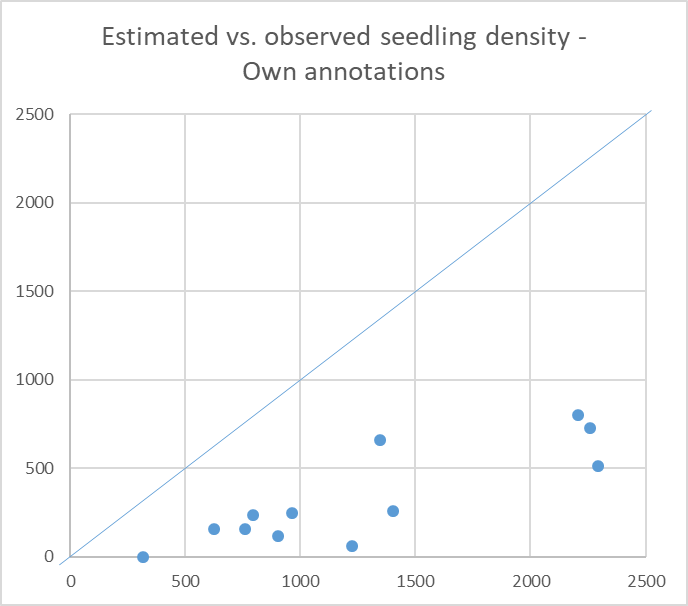


Figure 2 – Estimated vs. observed seedling density in the different AOIs (Areas of Interest). The blue line represents the 1:1 line, where estimated and observed values are equal.

# Discussion

* Model fitting

In principle, the models were tested changing 3 hyper parameters: image size, model size and batch size. Due to limited GPU memory, some combinations of those hyper parameters were not feasible. For instance, the GPU ran out of memory when using image size of 1000x1000 pixels and models bigger than n. In addition, increasing the batch size to also made the GPU run out of memory, regardless of the model size used.

The batch size did not seem to significantly influence the training speed. On the other hand, increasing the model size from n to m almost doubled the total training time. The same happened when increasing the model size from m to x.

The model trained on the full dataset performed better than the one trained on with own annotations. The own annotation dataset was composed by 28 images and some of the images did not have neither seedlings nor trees, configuring a rather limited source of training data for YOLO and, even though most ML metrics had higher values when training with the own annotation, the model performed poorly when tested in unseen data.

* Model performance

Regardless of the training data used, both models had relatively high RMSE and bias values when checked against field-measured tree locations. Such low accuracy could have been caused by a combination of different factors. First, figure 3 shows a situation where the bounding boxes drawn by the model do not englobe the whole seedling, but only a part of it. Such error could cause the reference and predicted bounding boxes not overlaying and, consequently, a potential omission error. Potentially, such error could be addressed by checking the annotations made on the training data to make sure a considerable number of big seedlings (such as the one highlighted in figure 3) are included in the training data and properly annotated.

Second, figure 4 shows a situation where big trees are not at all detected, contributing to the large negative bias observed when testing both models. Once more, more annotations on big large individuals could solve the issue and reduce the negative bias in trees/ha.

Finally, figure 5 shows two fallen trees in which green branches have been counted as seedlings. Only adding more annotations of tree seedlings might not be enough to avoid such errors, since the branches of a fallen tree might often resemble a seedling. In this case, a possible alternative to avoid such commission errors would be add a “fallen tree” class to the seedling detector. Thus, branches of such trees could be classified as a fallen tree and not as a seedling.



Figure 3 - Bounding box does not find whole trees, but parts of it

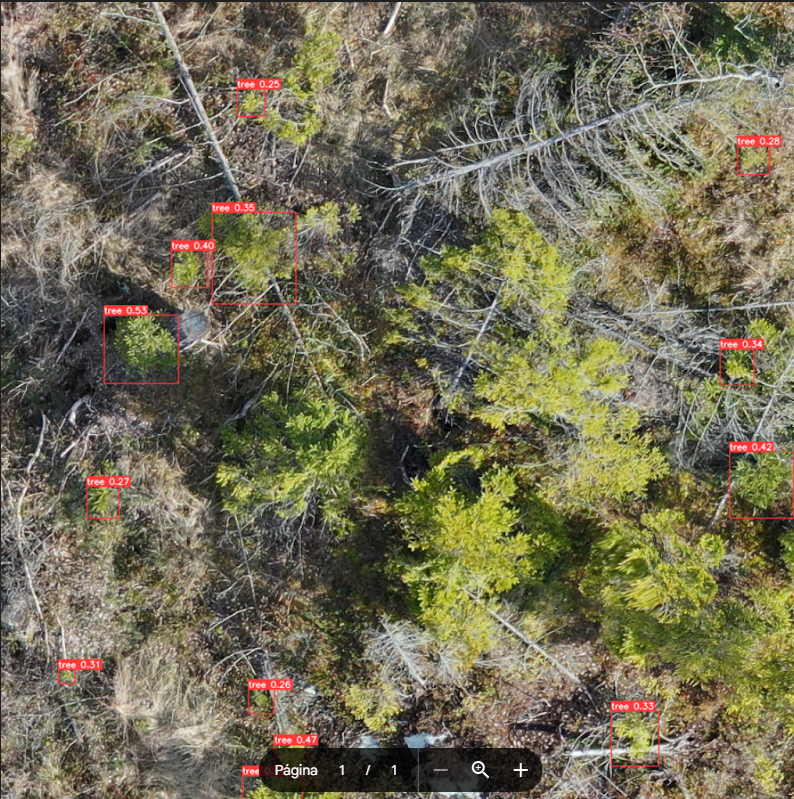


Figure 4 - Big trees are not detected



Figure 5 - Branches of felled trees detected as seedlings

# Upload your scripts to a github repo

Following Nico’s tutorial, create a github repo and upload your code there and paste the link to your github repo here.