NOVA course on deep learning in remote sensing

Home exercise

Raul de Paula Pires

# Objective

The objective of this assignment is to compare the effects of different annotation datasets for training a seedling detector, while also exploring various model sizes and hyperparameter configurations. For the purpose of seedling detection, we have employed the YOLOv8 model.

The annotation datasets used in this study differ in terms of the annotation process. One dataset consists of a smaller number of annotations made by a single person, while the other dataset contains a larger number of annotations made by a group of people. This variation allows us to investigate the impact of annotation quantity and diversity on the performance of the seedling detector.

By comparing these annotation datasets, we can assess the influence of the number of annotations and the diversity of annotators on the overall effectiveness of the trained seedling detector. Additionally, we will explore different model sizes and hyperparameter configurations to determine their impact on seedling detection accuracy. This assignment aims to provide insights into the effects of annotation dataset characteristics and model configurations on the performance of seedling detection systems.

# Data

## Train/val data

* + **Own Annotations**: 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). 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 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%) |

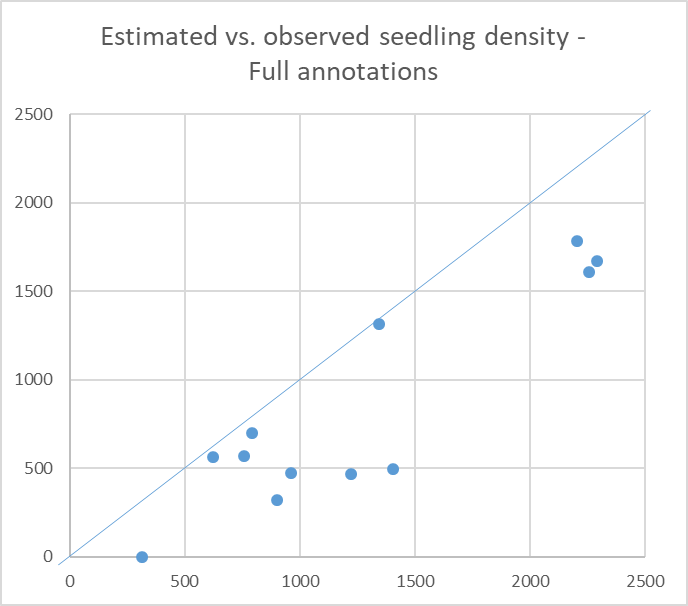
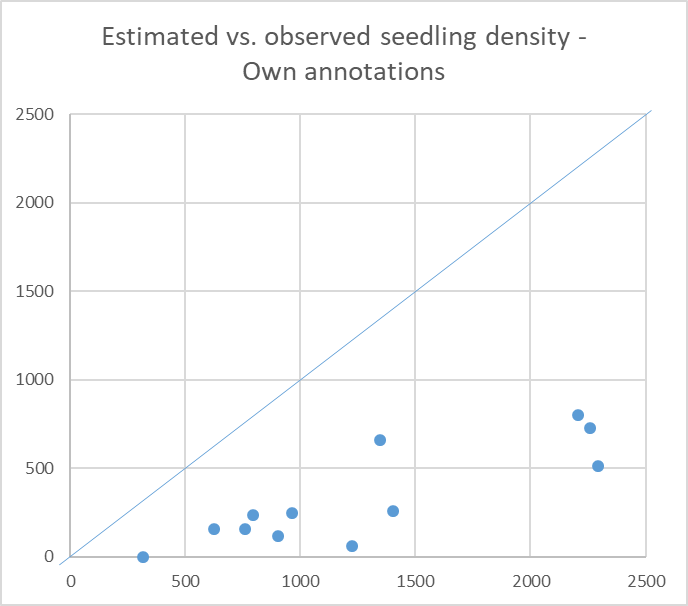


Figure 2 – Estimated vs. observed seedling density in the different AOIs (Areas of Interest). The blue line represents the 1:1 line, where estimated and observed values are equal.

# Discussion

* Model fitting

In practice, the models were evaluated by altering three hyperparameters: image size, model size, and batch size. However, due to limited GPU memory, certain combinations of these hyperparameters were not feasible. For instance, employing an image size of 1000x1000 pixels and models larger than a certain value caused the GPU to run out of memory. Similarly, increasing the batch size also resulted in GPU memory exhaustion, regardless of the model size used.

It was observed that changing the batch size did not significantly impact the training speed. However, increasing the model size from one value (n) to another (m) nearly doubled the overall training time. This trend persisted when further increasing the model size from m to x.

Furthermore, the model trained on the full dataset outperformed the one trained only on the own annotations. The own annotation dataset consisted of only 28 images, some of which did not contain any seedlings or trees. Although certain machine learning metrics exhibited higher values when training with the custom annotations, the model's performance was poor when evaluated on unseen data.

Altogether, these facts highlight the impact of dataset size and quality on the effectiveness of the trained models. The limited and less representative own annotation dataset led to poor performance, which highlights the importance of utilizing comprehensive and diverse training data.

* Model performance

Regardless of the training data used, both models exhibited relatively high root mean square error (RMSE) and bias values when compared against field-measured tree locations. These inaccuracies can be attributed to a combination of factors. In Figure 3, it is evident that the bounding boxes drawn by the model do not encompass the entire seedling, but only a portion of it. This error leads to misalignment between the reference and predicted bounding boxes, resulting in potential omission errors. To address this issue, it is crucial to thoroughly examine the annotations in the training data and ensure an adequate representation of larger seedlings, such as the one highlighted in Figure 3, to improve training accuracy.

In Figure 4, the models fail to detect larger trees, leading to a significant negative bias observed during testing. Increasing the number of annotations for larger individuals could help resolve this problem and reduce the negative bias in trees per hectare calculations.

Furthermore, Figure 5 demonstrates an instance where fallen trees with green branches are erroneously counted as seedlings. Simply augmenting the annotations for tree seedlings may not be sufficient to mitigate such errors, as the branches of fallen trees often resemble actual seedlings. In this case, a potential solution to minimize commission errors would be to introduce a "fallen tree" class to the seedling detector. This additional class would enable the classification of branches from fallen trees as fallen trees rather than seedlings.

Deep learning-based image detection techniques have their limitations, as evidenced by the high RMSE, bias values, and specific shortcomings observed in this study. However, by addressing these limitations, improvements can be achieved. Ensuring comprehensive and diverse training datasets, including annotations for larger seedlings and fallen trees, is crucial for enhancing the accuracy and robustness of such seedling detection model.



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

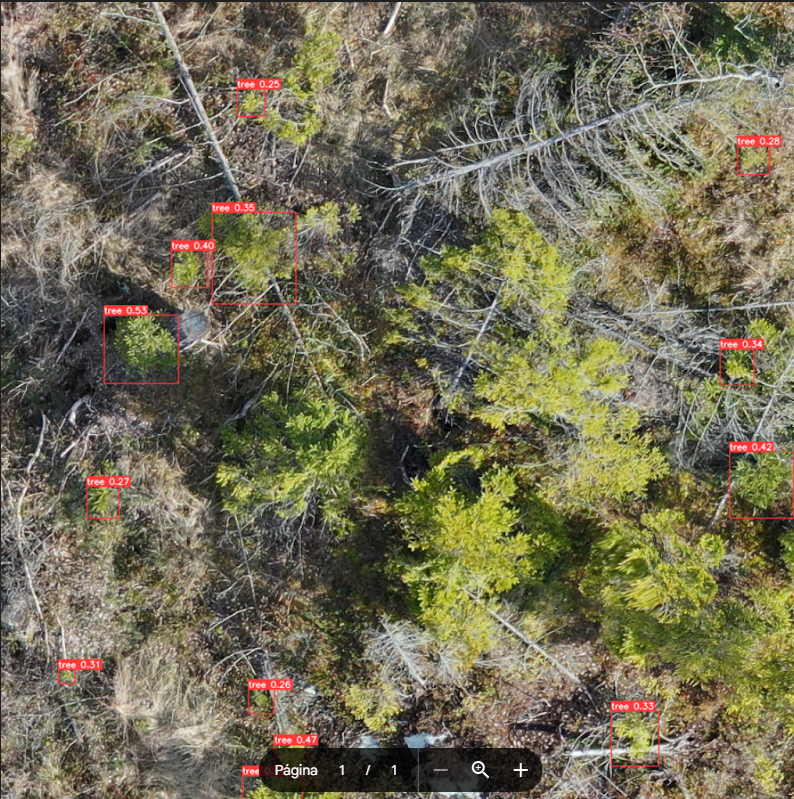


Figure 4 - Big trees are not detected



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.