Week 8

Mining Asset Detection (MAD)

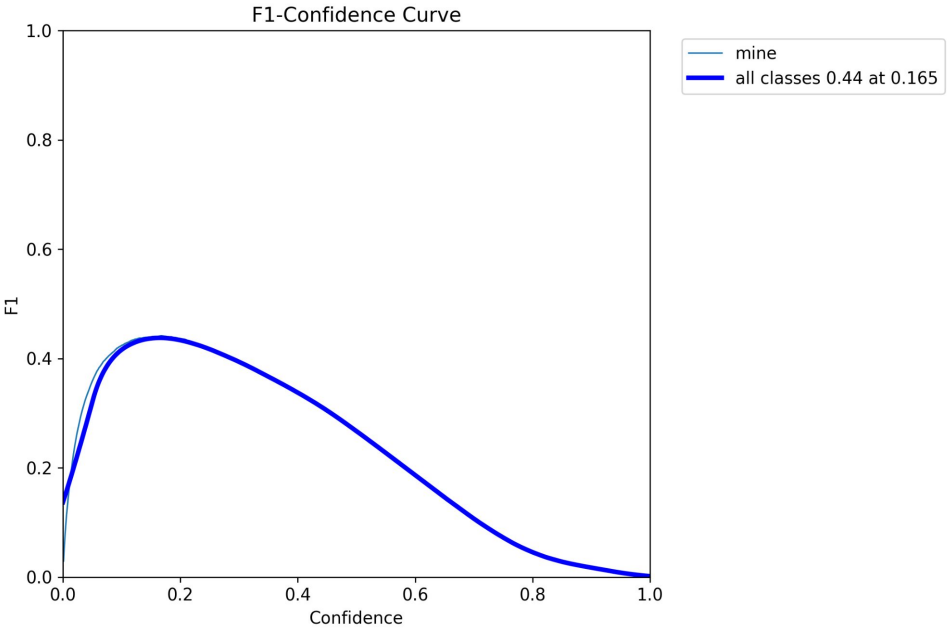
MAUS - Some details

- The creation of the MAUS dataset was done…
  - By hand
  - Around the perimeter of known excavation sites

=> If a whole mining area was not known, no mines within those will be labelled

=> Our contribution is essentially to find and fill these gaps

Let's talk about YOLO - Output metrics



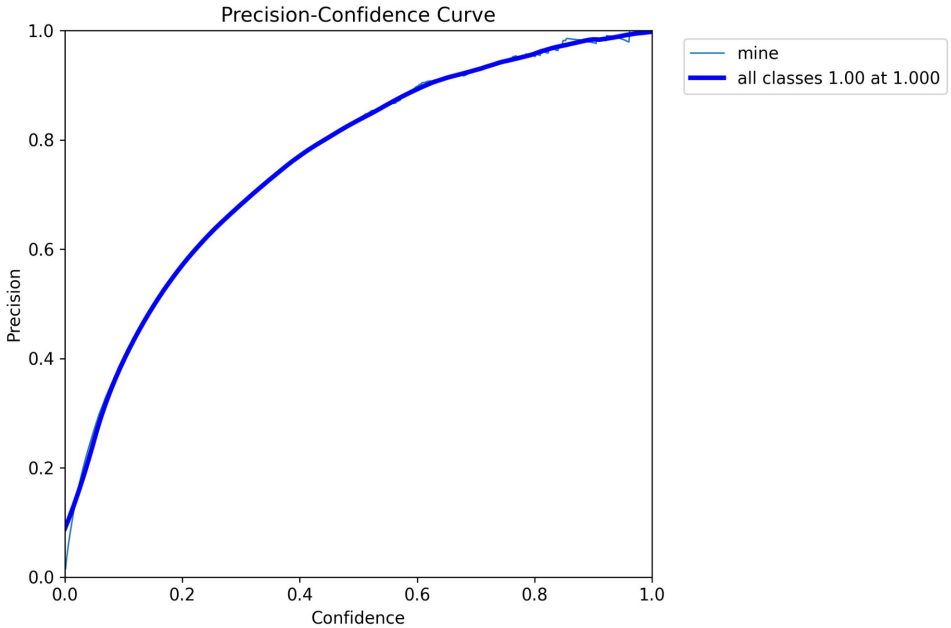
- F1-Score (Precision and Recall combined) vs. the prediction confidence.

In essence: YOLO outputs a confidence score for each classification, how well does the model perform if we use a certain threshold.

=> The **higher** the peak the better

=> The x-value of the highest peak is chosen

Let's talk about YOLO - Output metrics

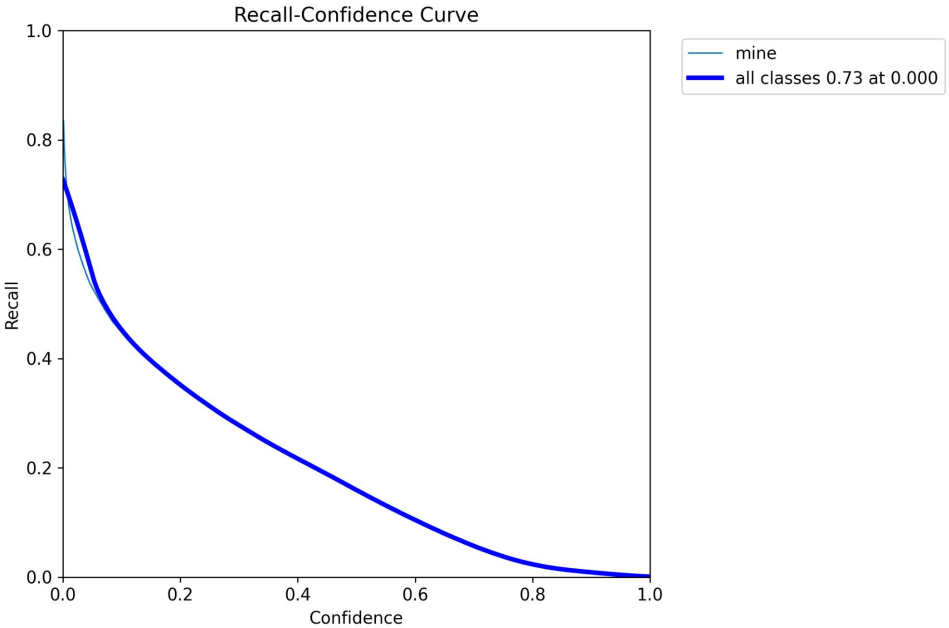


- Only precision for different confidence thresholds

In essence: How right is the model, if it predicts a mine, is there a mine?

We want already high precision for low confidence values in a good model.

Let's talk about YOLO - Output metrics

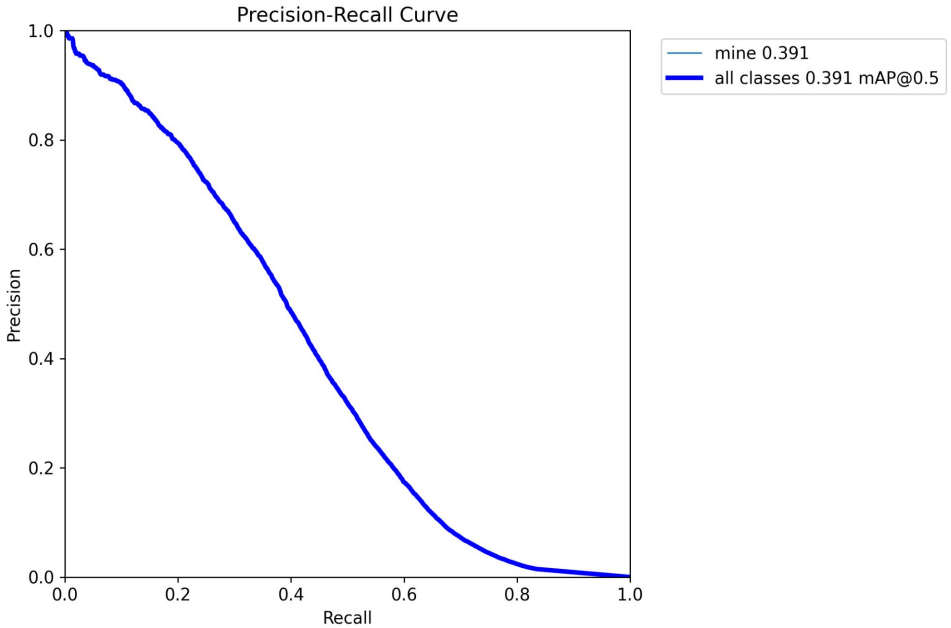


- Recall for different confidence thresholds

In essence: How many mines does the model catch

=> Curve should be high across all x

Let's talk about YOLO - Output metrics

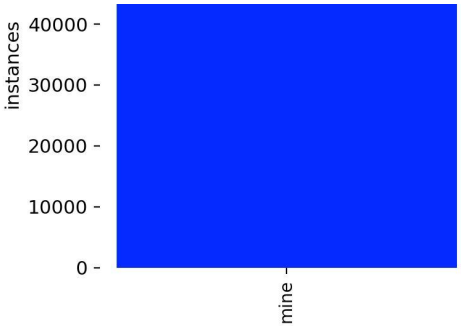


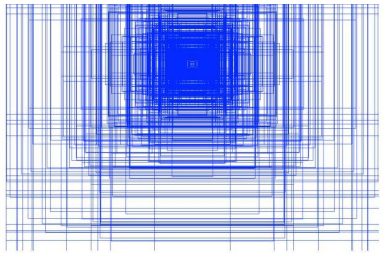
- Relation between precision and recall for different thresholds

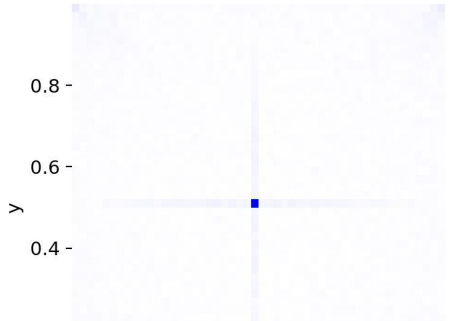
Goal is to have a high recall for high precisions

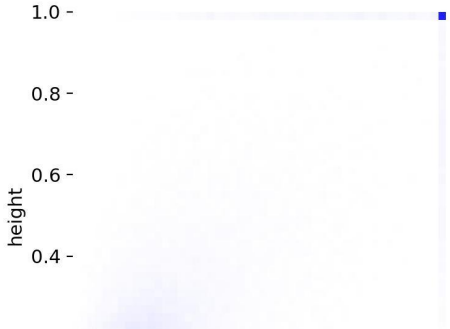
=> Curve should be close to upper right corner.

Let's talk about YOLO - Output metrics







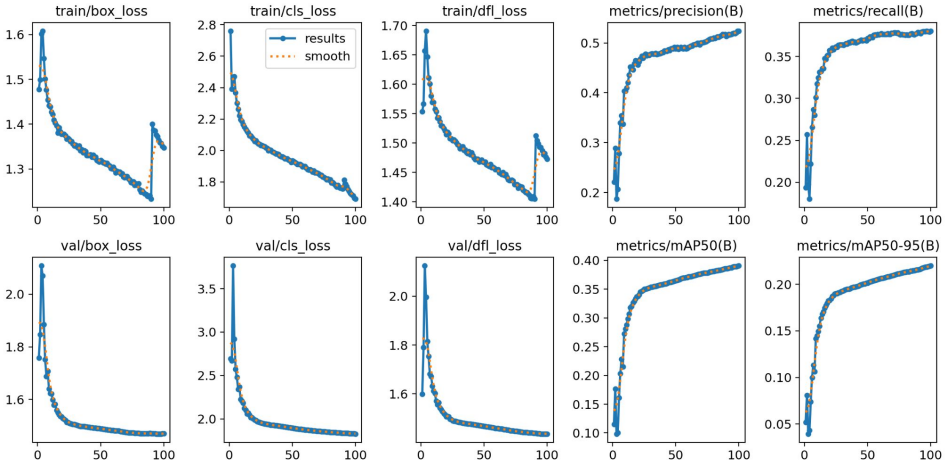


A significant amount of mines stretches across whole image:

- Concentration of centered bounding boxes
- Concentration of big bounding boxes

=> We can probably improve distribution by zooming out (Perturbation)

Let's talk about YOLO - Output metrics



- Box loss: How well the boxes match the ground truth (essentially 1 - intersection %)
- Class loss: Measures if correct class was predicted, in our case essentially if a mine was detected.
- Distribution Focal loss: How well the box is positioned (Correlated to box loss)

Foundation model -CLIP

GitHub

- Older but open source model of OpenAI
- Pretrained on Resnet (Probably not great for satellite images)
- Available as pytorch model (Should be cheap to run)

Foundation model -BLIP/Lavis

GitHub

- Anything related to LAnguage and VISion
  - Generation
  - Image to text
- Available as pytorch model (Should be cheap to run)
- Probably the best choice?

Foundation model -Segment Anything (SAM) 2.0

GitHub

- Just segmentation
- But often coupled with other models:
  - Segment first => Give the segments as input for visual model

Thoughts on the MAD interface

`mad grid create`

=> Results in a grid that defines satellite images

`mad grid download`

=> Download the provided grid

`mad yolo create --filter ... --extend ...`

=> Based on the given .tif images create a dataset that:

1. Filters images (e.g. takes out snowy tiles, low quality images.
2. Extends the resulting tiles by applying perturbations on the images like discussed.

`mad yolo train ????`

=> Train a model based on the defined yolo dataset and generate output with the final model being tested on a defined set of images.

Results from second test run

Even worse than the first :(

- Taking the averages will have sorted out some of the noise
  - Which we will have to do ourselves now
- Some satellite images look okay, some look horrible
  - Projection issues
  - Wrong sizes
  - Low fidelity
- Probable fix: Next stage of using larger GEE images and then doing cropping within the image ourselves