Model documentation and write-up

Information included in this section may be shared publicly with challenge results. You can respond to these questions in an e-mail or as an attached file. 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 am a senior data scientist at Nodalpoint Systems in Athens, Greece. I am a geologist and an oceanographer by education turned to data science through online courses and by taking part in numerous machine learning competitions.

2. What motivated you to compete in this challenge?
   I consider myself as a machine learning engineer who enjoys taking part in various machine learning competitions. The fact that this was a NASA competition and results could someday be used in space was an extra motivation to me.

3. High level summary of your approach: what did you do and why?
   Used a small detection model (yolo8n) to detect the satellite (using dataset of detection Track). Around it's center a rectangular of 384 pixels side is cropped and then a Siamese model with EfficientNetB0 backbone is trained using both the first image and one more with corresponding targets. In my solution only the 1st (= x) out of 7 targets was estimated and all others was set to 0. As a post processing step, all predictions are a weighted average with the previous in series prediction (pred=0.9*pred + 0.1*previous_pred) for every satellite series. Finally all predictions are clipped between (0, 400) and powered with 1.02 (preds=np.clip(preds,0,400)**1.02).

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

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

```
from ultralytics import YOLO
datayml = 'spaceship.yaml'
model = YOLO('yolov8n.pt')
results = model.train(data=datayml,
            imgsz=640,
            epochs=3,
            batch=32,
            device=[0],
            project ="finetune",
            plots=True)
```

Train a detection model for locating the satellite within the larger image.

```
def sat_center_crop(img, center=(512, 640), SIZE=384):
    y, x = img.shape[:2]
    halfS=int(SIZE/2)
    y1 = min(y-384, max(0, center[0]-halfS))
    y2 = min(y, max(384, center[0]+halfS))
    x1 = min(x-384, max(0, center[1]-halfS))
    x2 = min(x, max(384, center[1]+halfS))
    return img[y1:y2,x1:x2,:]
```

Based on the detection model, center crop an image to 384 * 384 final size.

```
if sat==sat0: # same satellite
    pred= 0.9*pred + 0.1*pred0
    pred0=pred
else:
    sat0=sat
    pred0=pred
```

Smooth predictions. Within the same satellite series, prediction is a weighted average with it's previous prediction.

6. Please provide the machine specs and time you used to run your model.
    - CPU (model): **Intel(R) Xeon(R) CPU X5650 @ 2.67GHz**
    - GPU (model or N/A): **GeForce GTX 1080**
    - Memory (GB): **70 GB**
    - OS: **Ubuntu 18.04**
    - Train duration: about **4 days** on local machine (can be optimized to run in about 1 day).
    - Inference duration: **less than half an hour on CPU**

7. 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

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?
   No

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

-

10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?
    Tried to predict all first 3 x, y, z targets but whether score was improved for my local validation this was not the case for the public and private LB.

11. 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?
    I would try other than Siamese network methodologies.

12. What simplifications could be made to run your solution faster without sacrificing significant accuracy?
    For model training I would separate the backbone from the rest of the model. As backbone layers are frozen, features from the backbone should be calculated once and not on every epoch. This was the case for my latest submissions but as also contained predictions for y and z targets, didn't score well on the private LB.