29.07- Genereal Research and Presentation

github base repo: https://gitlab.bewelltech.com.tr/Intern4/uav-building-detection

New dataset: https://www.sciencedirect.com/science/article/pii/S1569843222001595

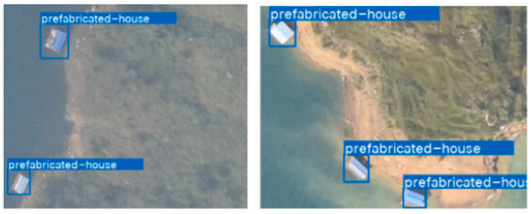
A context-scale-aware detector and a new benchmark for remote sensing small weak object detection in unmanned aerial vehicle images

To achieve real-scenario small weak object detection in UAV images, we present a 10-category dataset, UAVOD-10, consisting of 844 images and 18,234 instances. These instances are labeled with horizontal bounding boxes (HBBs). The images' widths range from 1000 to 4800 pixels, and their resolutions are approximately 0.15 m. As shown in Table 1, the phenomenon of class imbalance exists in the dataset, wherein building and prefabricated house objects have many instances, and quarry, pool, and landslide objects are fewer. Scale differences in objects of the same or different categories are apparent. The width of the smallest building instance is 9 pixels, the biggest one's width is 312 pixels, and the mean width is 74.85 pixels. Other classes, such as quarry, pool, and landslide, have larger average widths and standard deviations. These problems of class imbalance and scale difference are consistent with real-world applications.

Table 1
Attributes of objects from different categories in the proposed dataset.

Category	Number	Object width (pixels)			
		Max	Min	Mean	Std
Building	12,990	312	9	74.85	28.1
Prefabricated house	2060	876	12	47.77	35.11
Ship	1019	105	8	34.18	14.48
Well	648	115	9	33.51	11.78
Cable tower	587	245	5	53.23	44.64
Vehicle	506	133	10	24.71	12.1
Cultivation mesh cage	258	498	22	176.61	95.11
Quarry	83	1204	73	406.72	259.89
Pool	43	1946	20	211.76	269.89
Landslide	40	2044	57	526.24	461.7
All instances	18,234	2044	5	70.26	57.66

building

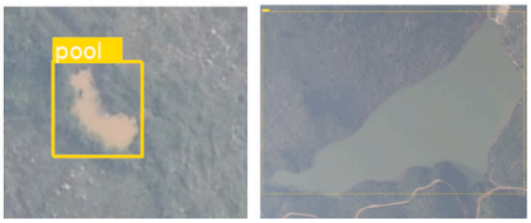


prefabricated-house

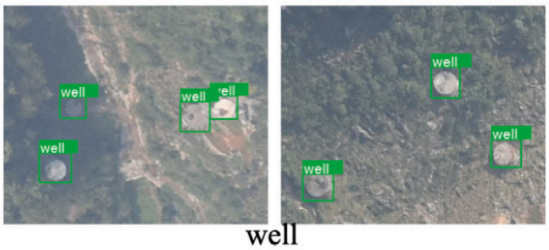
cable-tower

quarry

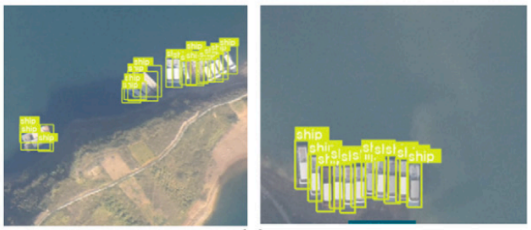
landslide



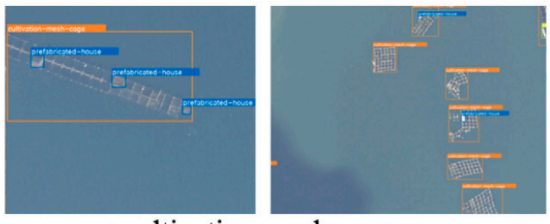
pool



well



ship



cultivation-mesh-cage

vehicle

New Tasks

- Tests on the new dataset (UAVOD-10). UAV images for detecting buildings. For UAV images
- Gathering the location information from images.
- Extend the tests on rescuenet dataset
- Multiclass classification for buildings (minor-medium damaged buildings)

T1- Write a clean and single script that digging data from image masks and converting to yolo format. For “rescuenet” and “uavod-10” datasets
T2 – Upload UAVOD-10 dataset to "Gdrive" for ease of access
T3- Comet-ML infrastructure base notebook for runs.

```
for obj in root.iter('object'):
            cls = obj.find('name').text
            if cls == "building":
                cls_id = 0
            elif cls == "prefabricated-house":
                cls_id = 1
            elif cls == "vehicle":
                cls_id = 2
            else:
                continue

            difficult = obj.find('difficult').text
            if int(difficult) == 1:
                continue
```

YOLOV8N VS YOLOV8S ON 30 EPOCHS AND 16 BATCH SIZE

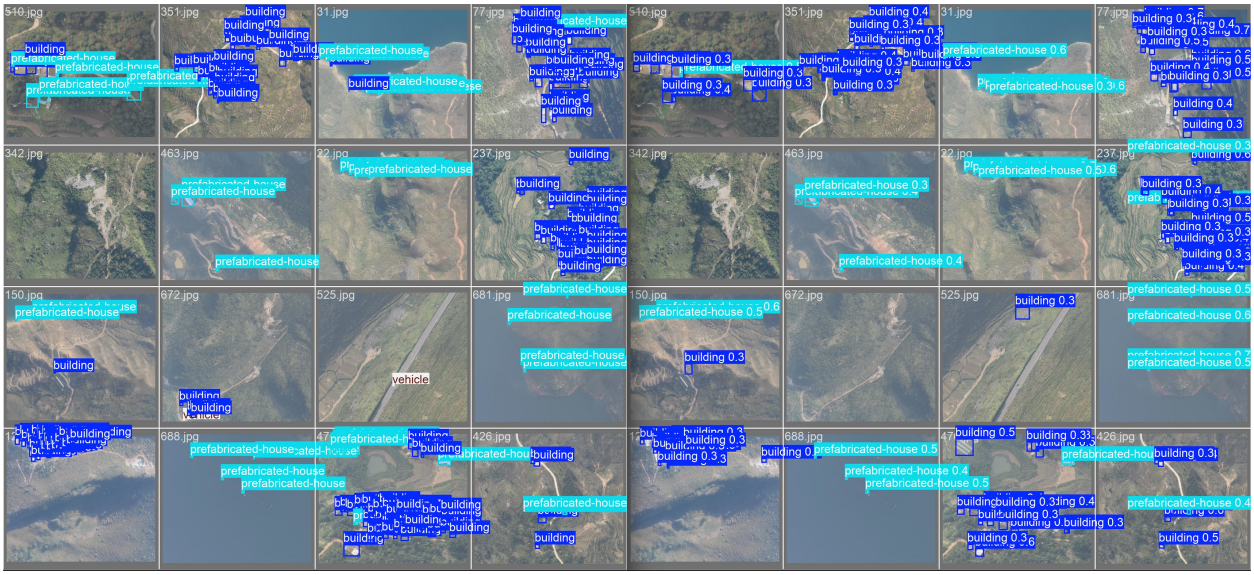
nano:

```
Validating building-detection-uav/train2/weights/best.pt...
Ultralytics YOLOv8.2.68 🚀 Python-3.10.12 torch-2.4.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|
                   all        169       2651      0.754      0.321      0.344      0.146
              building        126       2178      0.512      0.449      0.454      0.184
 prefabricated-house          83        391      0.751      0.516      0.577      0.251
               vehicle          24         82          1          0      0.002     0.0012
Speed: 0.5ms preprocess, 3.3ms inference, 0.0ms loss, 4.4ms postprocess per image
```

small:

```
Validating building-detection-uav/train4/weights/best.pt...
Ultralytics YOLOv8.2.68 🚀 Python-3.10.12 torch-2.4.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8s summary (fused): 168 layers, 11,126,745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 1
                   all        169       2651      0.781      0.365      0.395      0.188
              building        126       2178      0.624      0.513      0.562       0.26
 prefabricated-house          83        391      0.718      0.583      0.621      0.302
               vehicle          24         82          1          0    0.00325    0.00195
Speed: 1.0ms preprocess, 12.0ms inference, 0.0ms loss, 20.6ms postprocess per image
```

nano:



small:

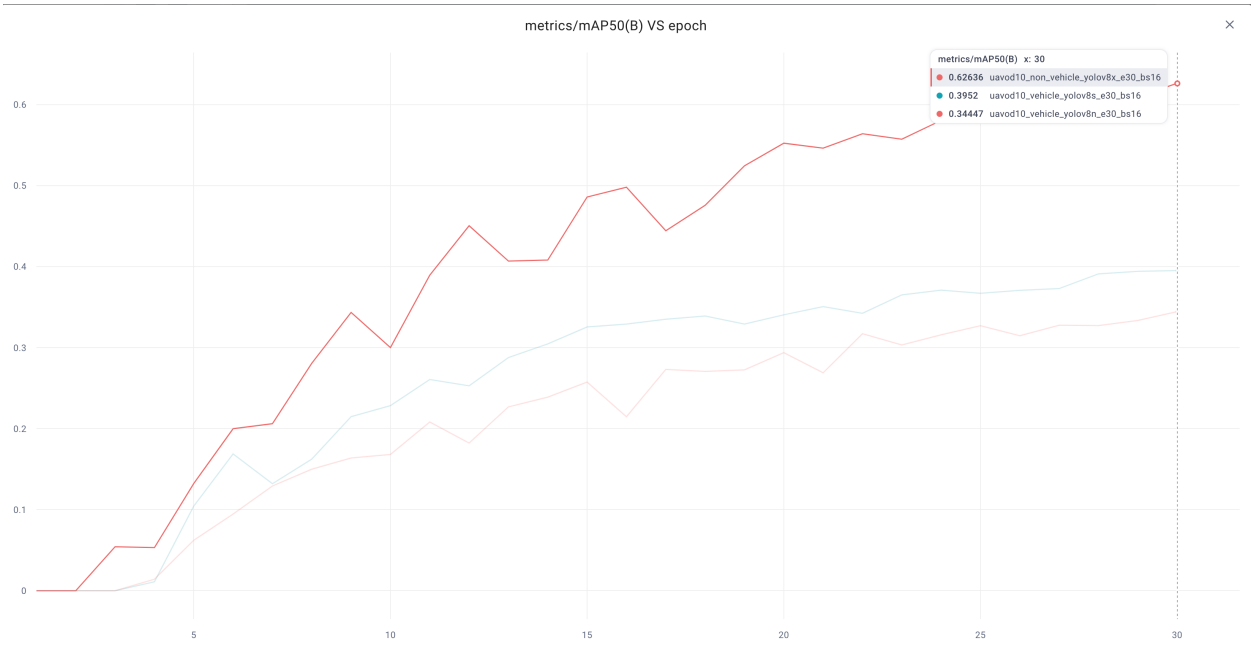
30.07- GOING ON NEW TASKS

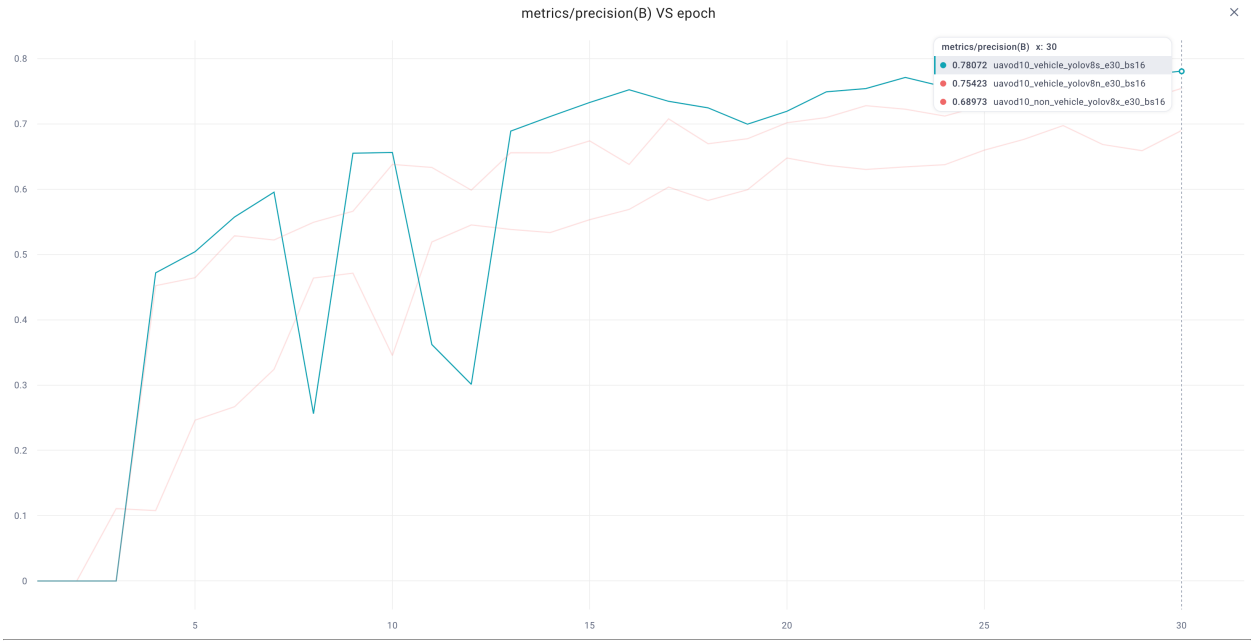
Vehicle class can be removed from labels. There are only a bunch of vehicles so it leads our case to an imbalance data problem.

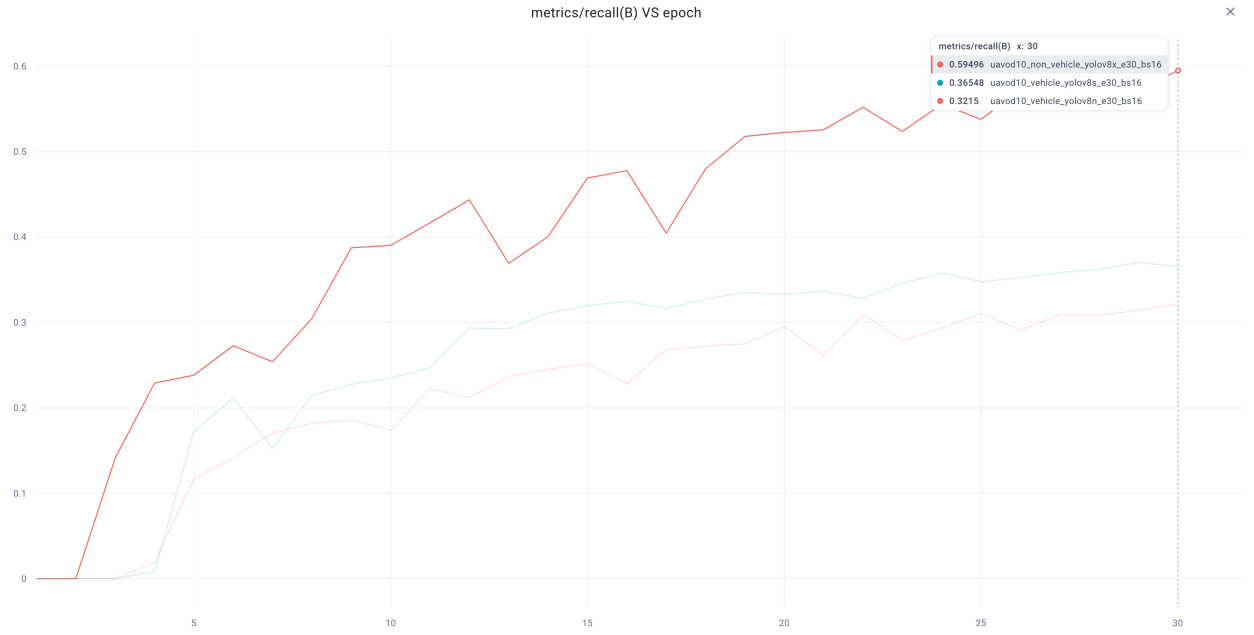
Tasks from yesterday
T1- Write a clean and single script that digging data from image masks and converting to yolo format. For “rescuenet” and “~~uavod-10~~” datasets
~~T2 - Upload UAVOD-10 dataset to "Gdrive" for ease of access~~
~~T3- Comet-ML infrastructure base notebook for runs.~~
T4 - Precise location detection
T5 - Vehicle class removal from labels and new tests

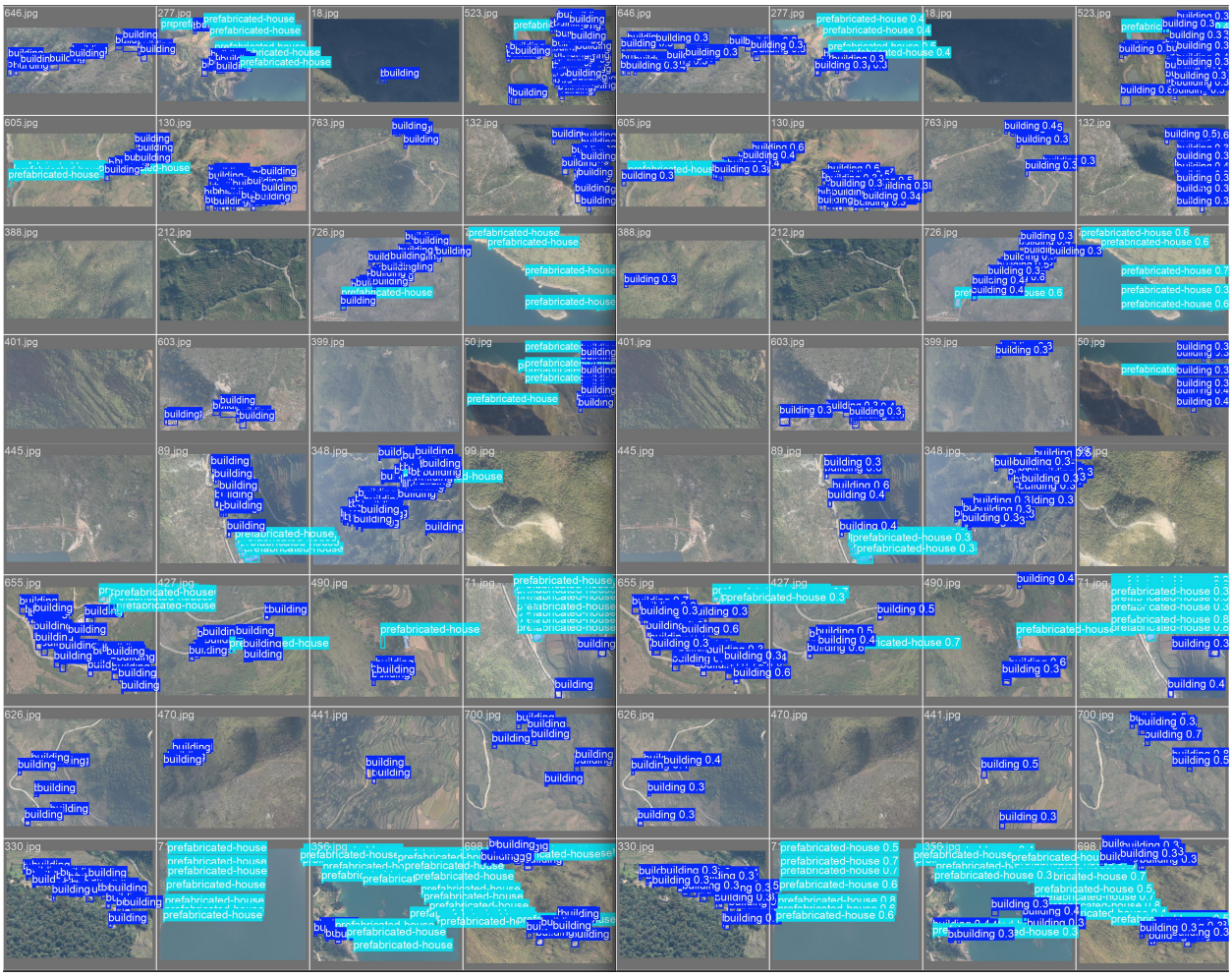
size problem on runs: save only best and last epoch config.
Check results at url: cometML portal
Model comparison YOLOV8[N-S-X]
yolov8x 30 epoch 16 batch size

```
Validating building-detection-uav/train/weights/best.pt...
Ultralytics YOLOv8.2.69 🚀 Python-3.10.12 torch-2.4.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8x summary (fused): 268 layers, 68,125,494 parameters, 0 gradients, 257.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 6/6 [00:18<00:00,  3.16s/it]
                   all        169       2534       0.69      0.595      0.626      0.315
              building        122       2160      0.611      0.596      0.595      0.295
   prefabricated-house         84        374      0.768      0.594      0.658      0.335
Speed: 0.5ms preprocess, 28.4ms inference, 0.0ms loss, 6.6ms postprocess per image
```




metrics/precision(B) VS epoch
metrics/precision(B) x: 30
0.78072 uavod10_vehicle_yolov8s_e30_bs16
0.75423 uavod10_vehicle_yolov8n_e30_bs16
0.68973 uavod10_non_vehicle_yolov8x_e30_bs16
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30


metrics/recall(B) VS epoch
metrics/recall(B) x: 30
0.59496 uavod10_non_vehicle_yolov8x_e30_bs16
0.36548 uavod10_vehicle_yolov8s_e30_bs16
0.3215 uavod10_vehicle_yolov8n_e30_bs16
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30



COCO

Model	Test Size	#Params	FLOPs	AP^{val}	Latency
YOLOv10-N	640	2.3M	6.7G	38.5%	1.84ms
YOLOv10-S	640	7.2M	21.6G	46.3%	2.49ms
YOLOv10-M	640	15.4M	59.1G	51.1%	4.74ms
YOLOv10-B	640	19.1M	92.0G	52.5%	5.74ms
YOLOv10-L	640	24.4M	120.3G	53.2%	7.28ms
YOLOv10-X	640	29.5M	160.4G	54.4%	10.70ms

YOLOV10 TESTS- N-S-M

N:

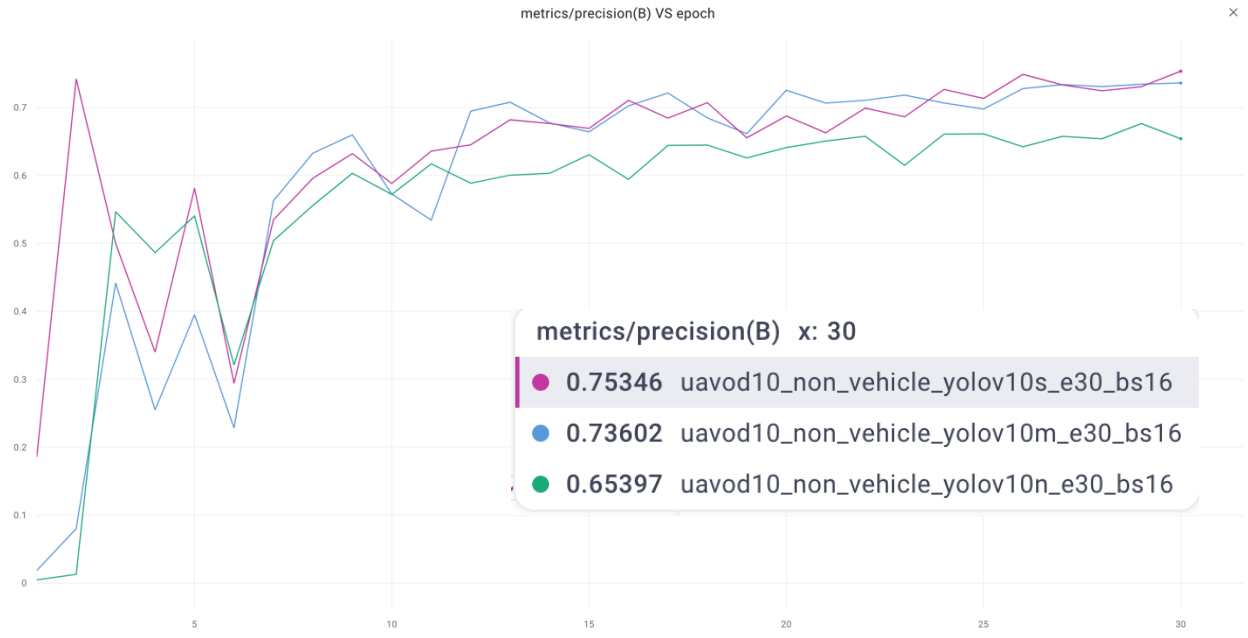
```
Validating building-detection-uav/train/weights/best.pt...
Ultralytics YOLOv8.1.34 🚀 Python-3.10.12 torch-2.3.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv10n summary (fused): 285 layers, 2695196 parameters, 0 gradients, 8.2 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 6/6 [00:13<00:00,  2.22s/it]
                   all        169       2534      0.654      0.659      0.671      0.367
              building        169       2160      0.597      0.688       0.65      0.353
   prefabricated-house        169        374      0.711      0.631      0.691       0.38
Speed: 0.3ms preprocess, 6.1ms inference, 0.0ms loss, 0.1ms postprocess per image
Saving building-detection-uav/train/predictions.json...
Results saved to building-detection-uav/train
```

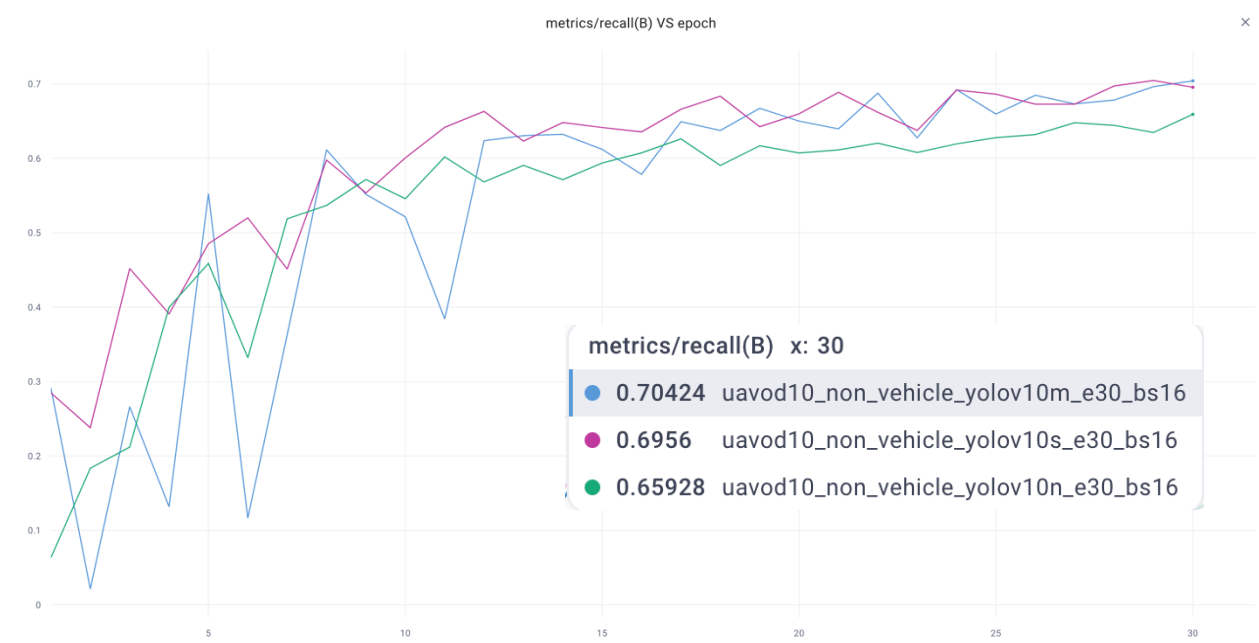
S:

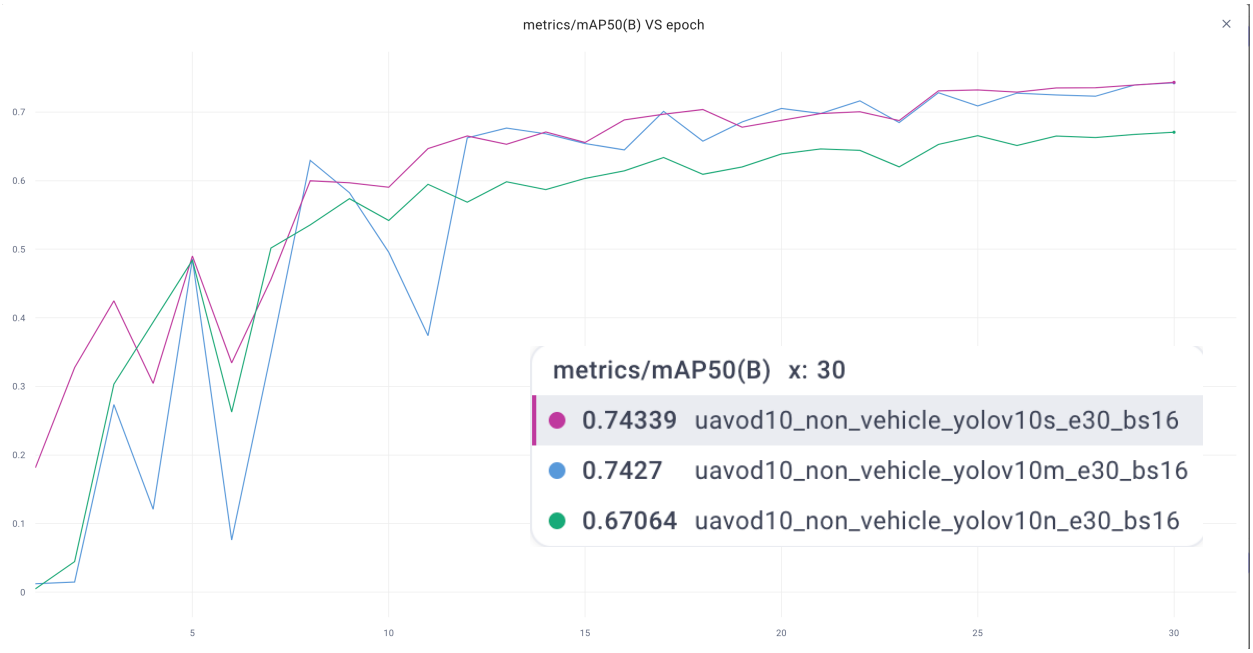
```
Validating building-detection-uav/train2/weights/best.pt...
Ultralytics YOLOv8.1.34 🚀 Python-3.10.12 torch-2.3.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv10s summary (fused): 293 layers, 8036508 parameters, 0 gradients, 24.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 6/6 [00:14<00:00,  2.47s/it]
                   all        169       2534      0.753      0.696      0.743      0.419
              building        169       2160      0.708        0.7      0.724      0.413
   prefabricated-house        169        374      0.799      0.691      0.763      0.426
Speed: 0.6ms preprocess, 12.6ms inference, 0.2ms loss, 0.3ms postprocess per image
Saving building-detection-uav/train2/predictions.json...
Results saved to building-detection-uav/train2
```

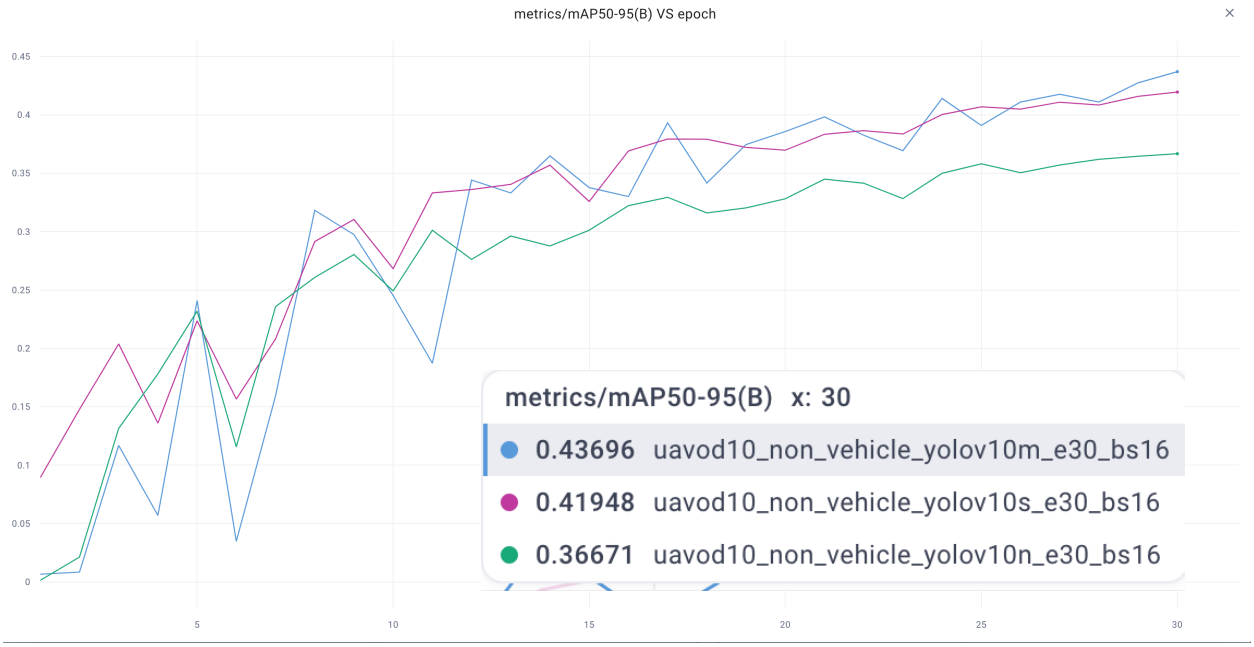
M:

```
Validating building-detection-uav/train/weights/best.pt...
Ultralytics YOLOv8.1.34 🚀 Python-3.10.12 torch-2.3.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv10m summary (fused): 369 layers, 16452700 parameters, 0 gradients, 63.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 6/6 [00:19<00:00,  3.22s/it]
                   all        169       2534      0.736      0.704      0.743      0.437
              building        169       2160       0.66      0.732      0.722      0.422
   prefabricated-house        169        374      0.812      0.676      0.764      0.452
Speed: 0.6ms preprocess, 28.9ms inference, 0.0ms loss, 0.2ms postprocess per image
Saving building-detection-uav/train/predictions.json...
Results saved to building-detection-uav/train
```



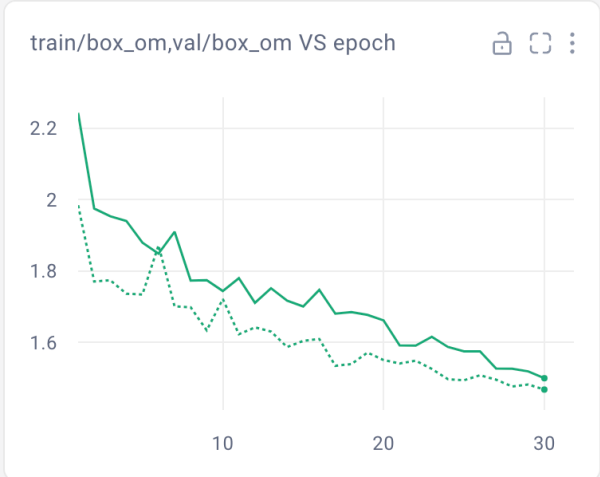

metrics/recall(B) VS epoch
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
metrics/recall(B) x: 30
0.70424 uavod10_non_vehicle_yolov10m_e30_bs16
0.6956 uavod10_non_vehicle_yolov10s_e30_bs16
0.65928 uavod10_non_vehicle_yolov10n_e30_bs16

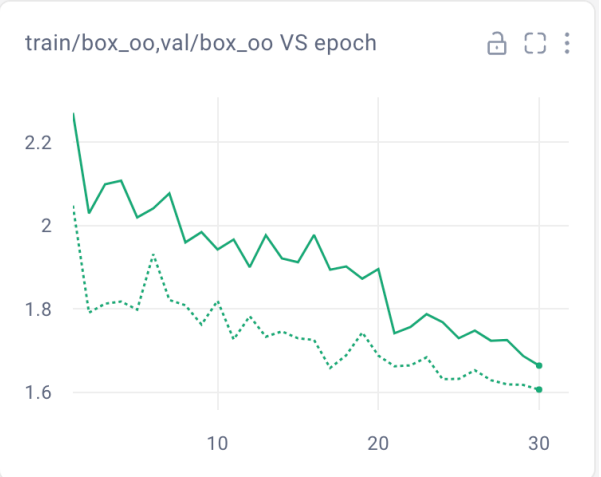

metrics/mAP50(B) VS epoch
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
metrics/mAP50(B) x: 30
0.74339 uavod10_non_vehicle_yolov10s_e30_bs16
0.7427 uavod10_non_vehicle_yolov10m_e30_bs16
0.67064 uavod10_non_vehicle_yolov10n_e30_bs16


metrics/mAP50-95(B) VS epoch
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
5
10
15
20
25
30
metrics/mAP50-95(B) x: 30
0.43696 uavod10_non_vehicle_yolov10m_e30_bs16
0.41948 uavod10_non_vehicle_yolov10s_e30_bs16
0.36671 uavod10_non_vehicle_yolov10n_e30_bs16

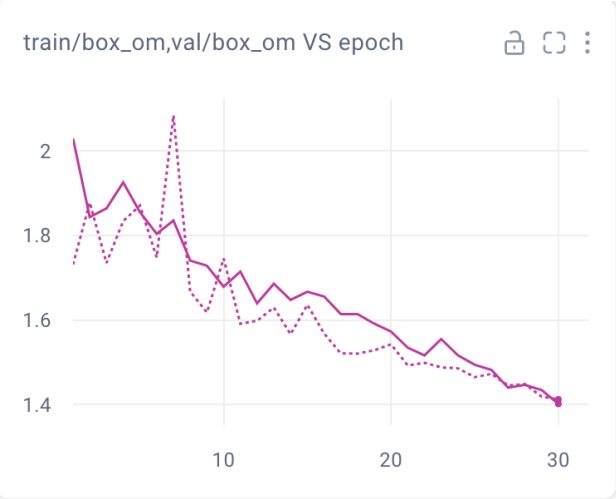
TRAIN VS VAL LOSS GRAPHS:

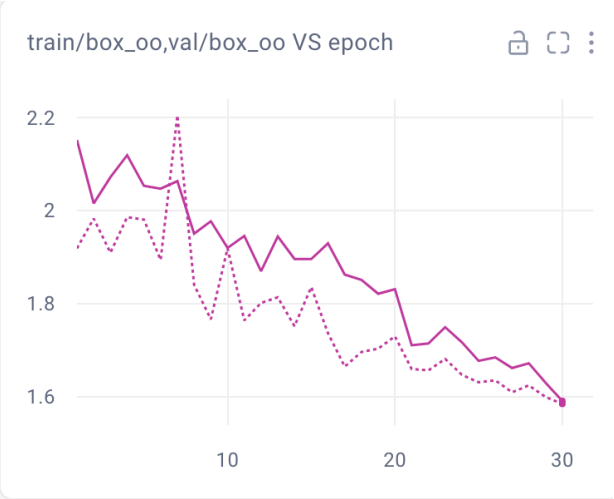
N:



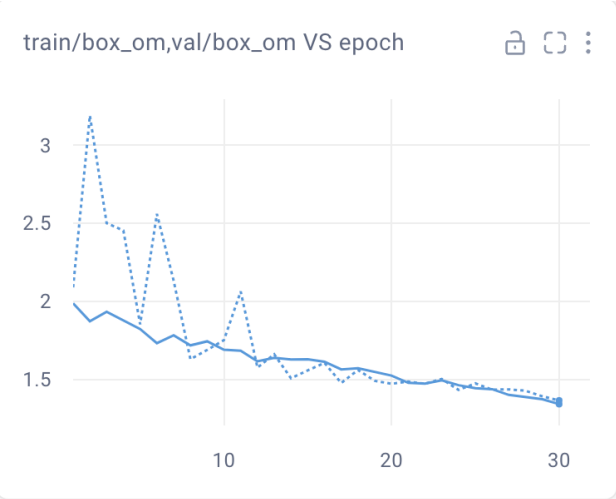


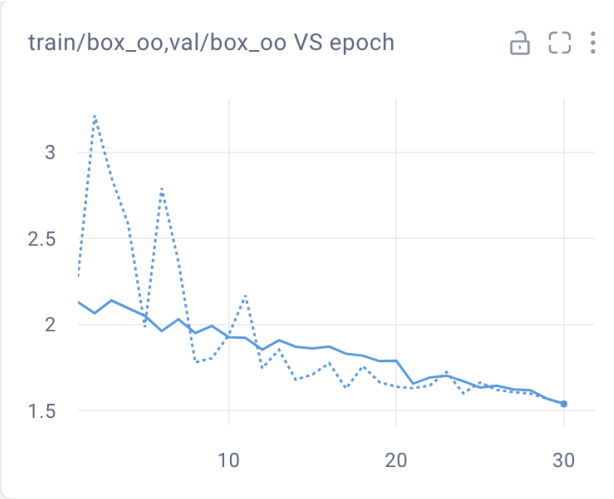
S:





M:





OUTPUTS:

N VS S VS M

N:

S:

M:

BEST YOLOV8 MODEL VS BEST YOLOV10 MODEL COMPARISON