Technische Universität Braunschweig

10-Sept-24

Evaluation of different YOLO models on DIOR dataset

Group 5

Vigneshwar Karuppiah Ramanathan

Amulya Harihara Narayana Rao

M.Sc. Computational Sciences in Engineering

Outline

1. **Introduction and motivation**
2. DIOR dataset
3. Regularization and Hyperparameters
4. Metrics
5. Models and Experimental Setup
6. Results
7. Summary and References

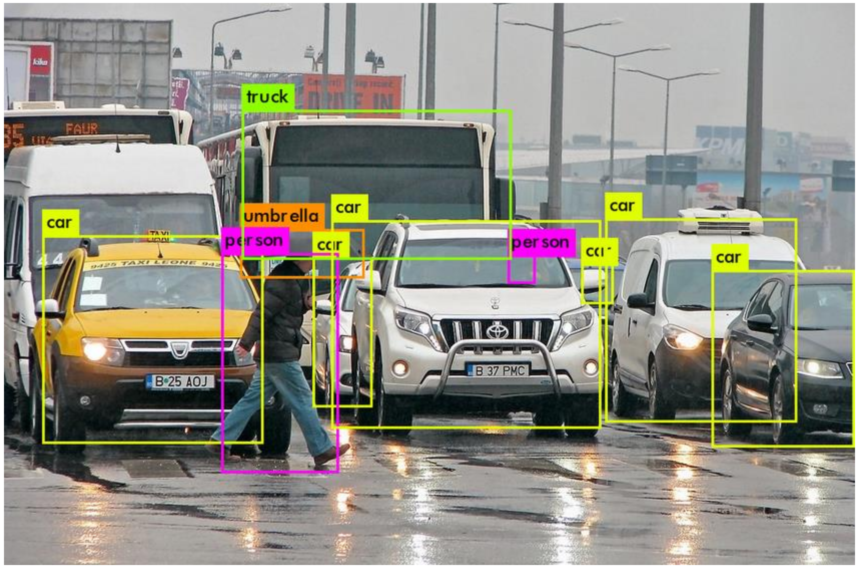
Introduction

Problem task

- Object detection in computer vision is the process of identifying and localizing objects within digital images or videos.

Motivation

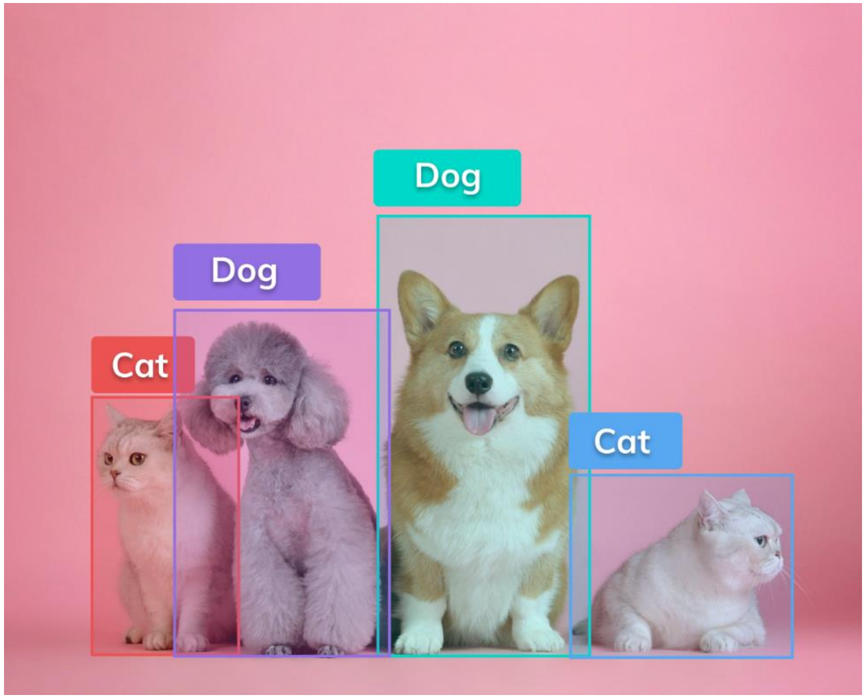
- Has many applications in remote sensing and various other fields
- Pixel-level detections can be accurately mapped to real-world locations
- Focus on real time application



Kedar Potdar et. al., "**A Convolutional Neural Network based Live Object Recognition System as Blind Aid", Computer Vision and Pattern Recognition (2018). DOI:** https://doi.org/10.48550/arXiv.1811.10399

Inputs and outputs of a object detection model

- **Inputs** are images or videos
- **Output** is bounding boxes with class labels
- **Regression problem** because the class probabilities as the output space is continuous in nature
- **Output representation** is done in following ways:
  - Top-left, Bottom-right coordinates
  - Top-left coordinate, width, height
  - Top-left coordinate, width, height and Theta
  - 'n' number of (x , y) coordinates



Source : **V7 Labs**, https://www.v7labs.com/blog/object-detection-guideabs

DIOR Dataset

Disadvantages of DIOR's predecessors and other publicly available dataset:

- Low numbers of images
- Small scale object categories
- Image diversity & variations are insufficient

Comparison between the proposed DIOR dataset and nine publicly available object detection datasets in earth observation community.

Datasets	# Categories	# Images	# Instances	Image width	Annotation way	Year
TAS	1	30	1319	792	horizontal bounding box	2008
SZTAKI-INRIA	1	9	665	~800	oriented bounding box	2012
NWPU VHR-10	10	800	3775	~1000	horizontal bounding box	2014
VEDAI	9	1210	3640	1024	oriented bounding box	2015
UCAS-AOD	2	910	6029	1280	horizontal bounding box	2015
DLR 3K Vehicle	2	20	14235	5616	oriented bounding box	2015
HRSC2016	1	1070	2976	~1000	oriented bounding box	2016
RSOD	4	976	6950	~1000	horizontal bounding box	2017
DOTA	15	2806	188282	800-4000	oriented bounding box	2017
DIOR (ours)	20	23463	192472	800	horizontal bounding box	2018

Ke Li *et. al.*, "**Object Detection in Optical Remote Sensing Images: A Survey and A New Benchmark**" *ISPRS journal of photogrammetry and remote sensing* (2020). DOI: 10.1016/j.isprsjprs.2019.11.023

Outline

1. Introduction and motivation
2. **DIOR dataset**
3. Regularization and Hyperparameters
4. Metrics
5. Models and Experimental Setup
6. Result
7. Summary and References

DIOR Dataset

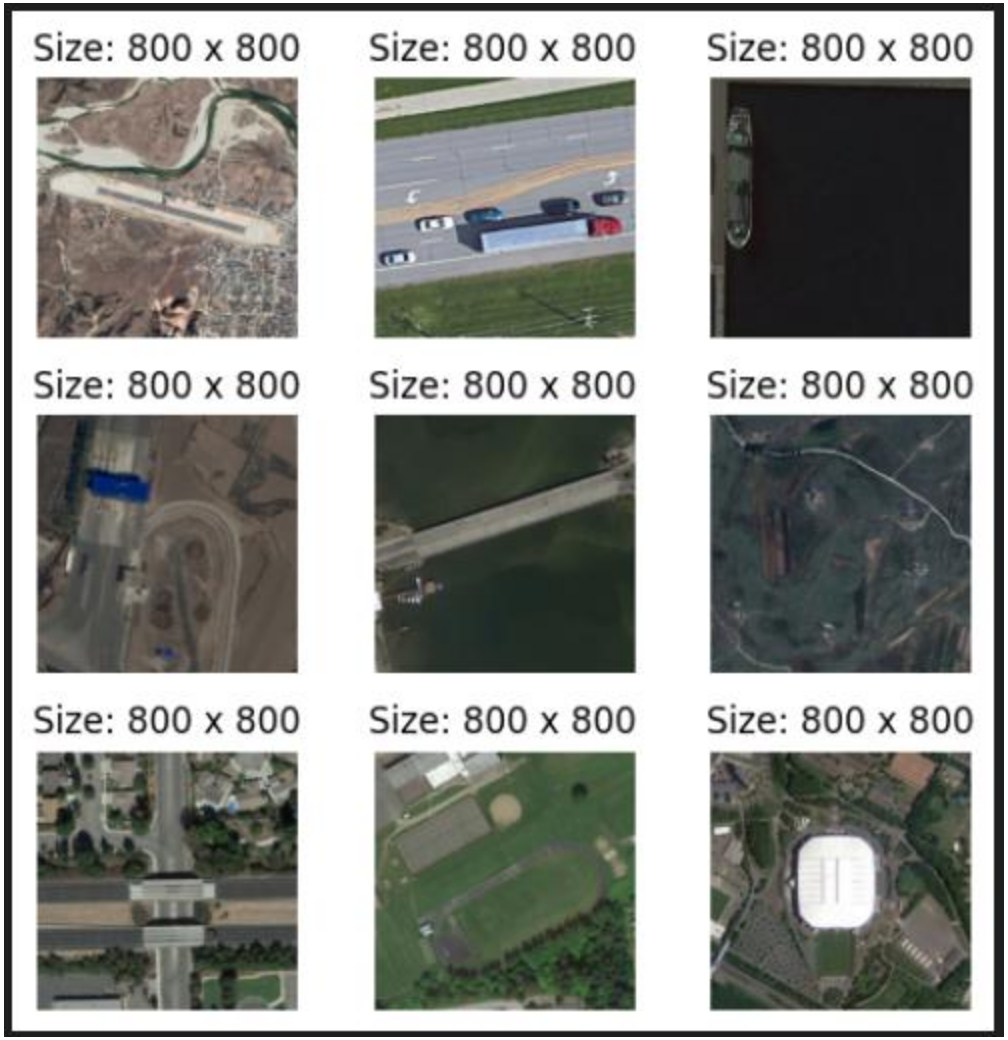
- Latest dataset out of the three-dataset provided
- Commonly used 10 categories were selected from DOTA and NWPU VHR-10
- Includes objects from suburban areas too
- Diversity of object size helpful in real world tasks

Characteristics of DIOR

1. Large scale
    - contains 23463 images and 192472 instances, covering 20 classes
    - each category contains about 1200 images
2. Large range of object size variations
3. Rich image variations
4. High inter-class similarity and intra-class diversity

Sample Images

- 800 x 800 → pixel images
- 3 Channels
- Spatial resolution 0.5 m to 30 m
- Richer variations in viewpoint, translation, illumination, background, object pose and appearance, occlusion, etc
- Total number of possible combination of classes is 526

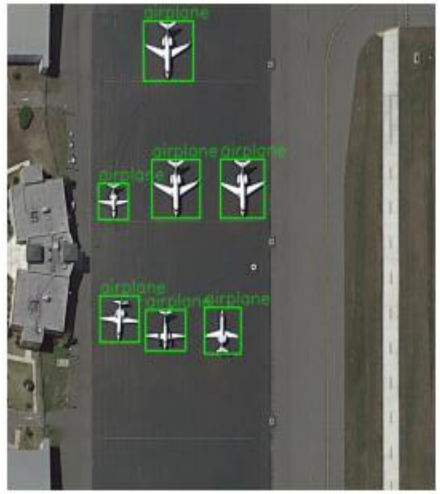


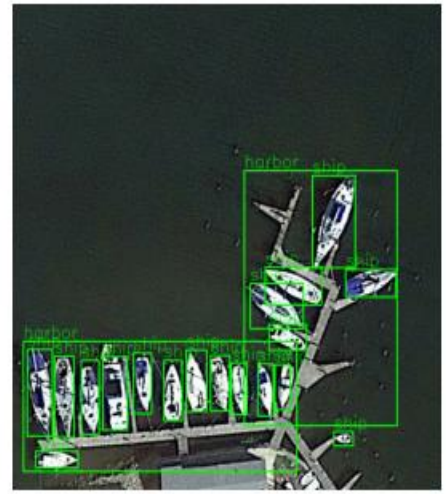
Ke Li *et. al.*, "**Object Detection in Optical Remote Sensing Images: A Survey and A New Benchmark"** *ISPRS journal of photogrammetry and remote sensing* (2020). DOI: 10.1016/j.isprsjprs.2019.11.023

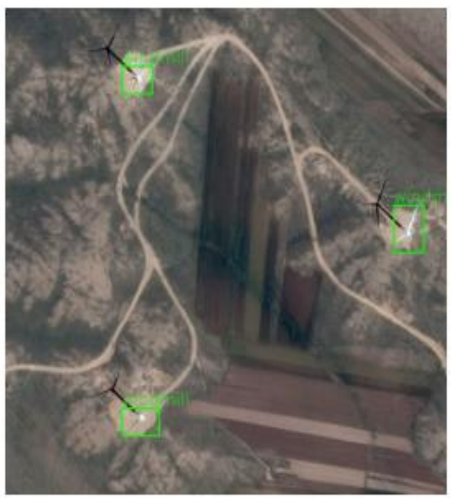
Classes available in DIOR Dataset (20 Classes)

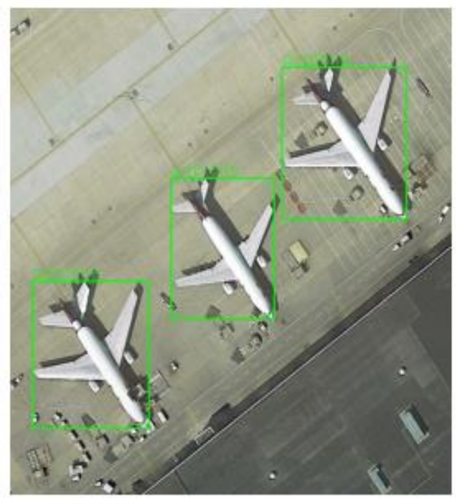
Ke Li *et. al.*, "**Object Detection in Optical Remote Sensing Images: A Survey and A New Benchmark"** *ISPRS journal of photogrammetry and remote sensing* (2020). DOI: 10.1016/j.isprsjprs.2019.11.023

Sample Images with bounding boxes and labels









General structure of DIOR dataset

Dataset

- Test dataset contains 11738
- Train Val dataset contains 11725

```
    |
    ├──Dior
    |    ├──Annotations
    |    |    └──Annotations
    |    |         ├──Horizontal Bounding Boxes
    |    |         |    └── 00001.xml
    |    |         |    └── 00002.xml
    |    |         |    └── .....
    |    |         └──Oriented Bounding Boxes
    |    |              └── 00001.xml
    |    |              └── 00002.xml
    |    |              └── .....
    |    ├──ImageSets
    |    |    ├──Layout
    |    |    ├──Main
    |    |    |    └── test.txt
    |    |    |    └── train.txt
    |    |    |    └── val.txt
    |    |    └──Segmentation
    |    ├──JPEGImages-test-002
    |    |    └──JPEGImages-test
    |    |    |    └── 11726.jpg
    |    |    |    └── .....
    |    |    |    └── 23463.jpg
    |    └──JPEGImages-trainval
    |    |    |    └── 00001.jpg
    |    |    |    └── .....
    |    |    |    └── 11725.jpg
```

Annotation Format

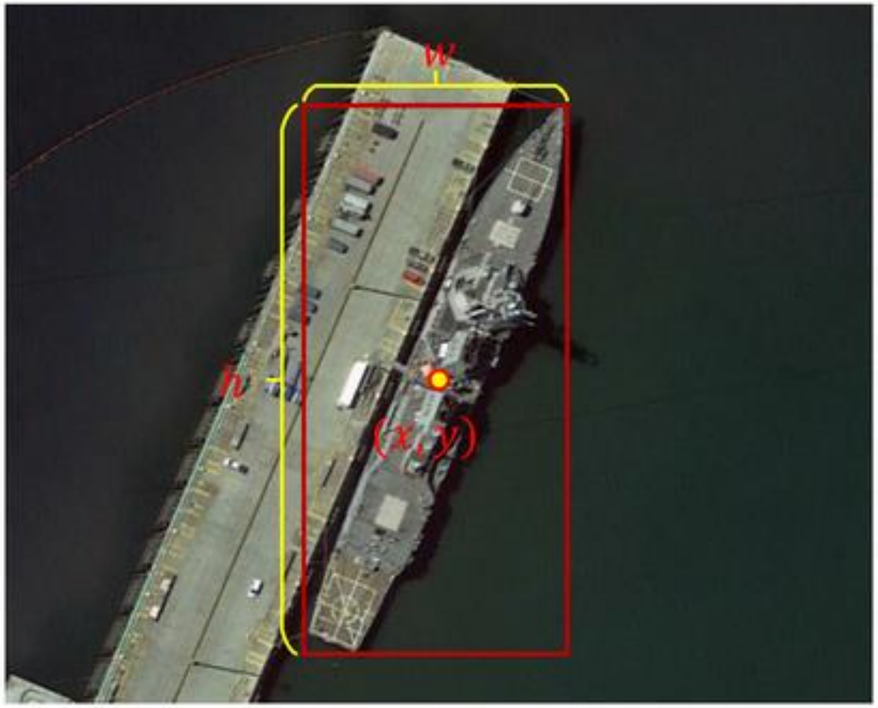
- Pascal VOC (XML File)

```xml
<annotation>
    <filename>00001.jpg</filename>
    <source>
        <database>DIOR</database>
    </source>
    <size>
        <width>800</width>
        <height>800</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>golffield</name>
        <pose>Unspecified</pose>
        <bndbox>
            <xmin>133</xmin>
            <ymin>237</ymin>
            <xmax>684</xmax>
            <ymax>672</ymax>
        </bndbox>
    </object>
</annotation>
```

Types of annotation output

The final object detection output can we visualized in two ways, namely:

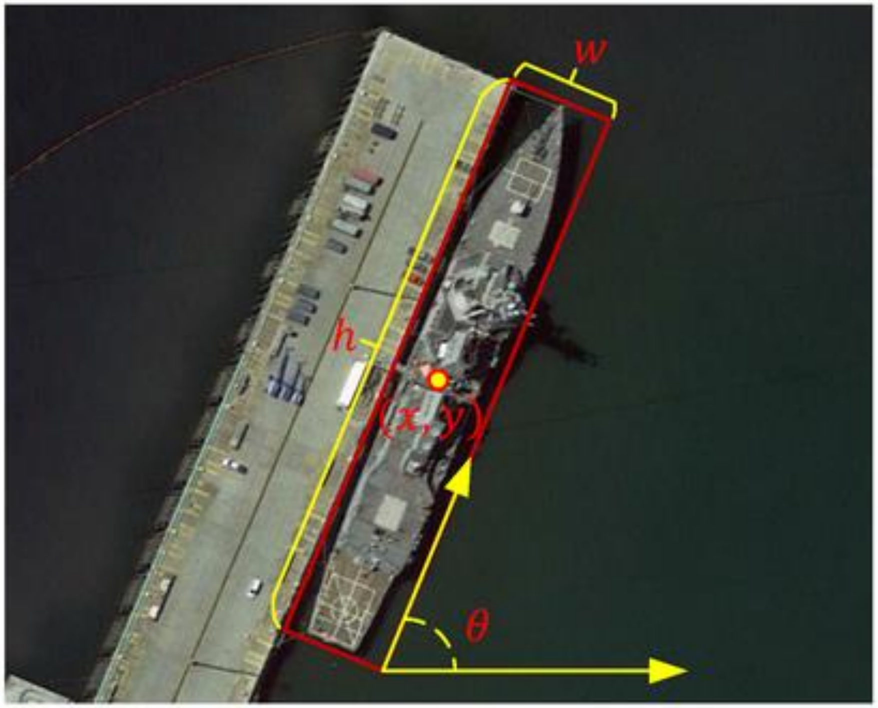
Horizontal Bounding Box



4 Parameters

(X_min, Y_min, X_max, Y_max)

Oriented Bounding Box

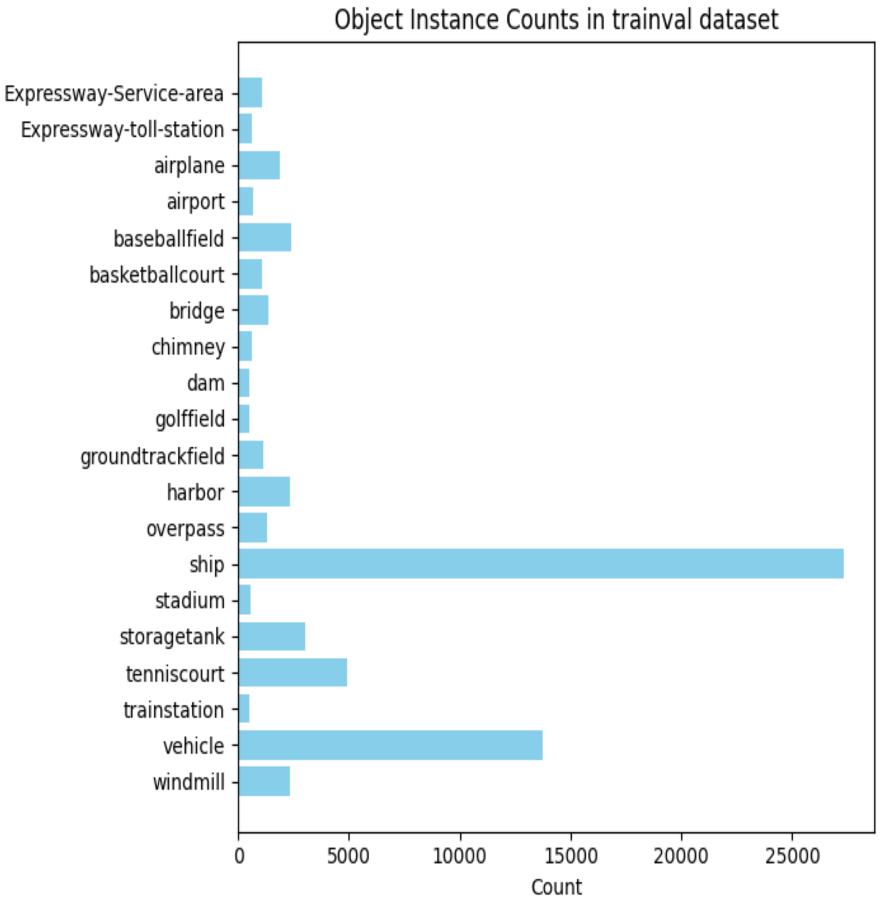


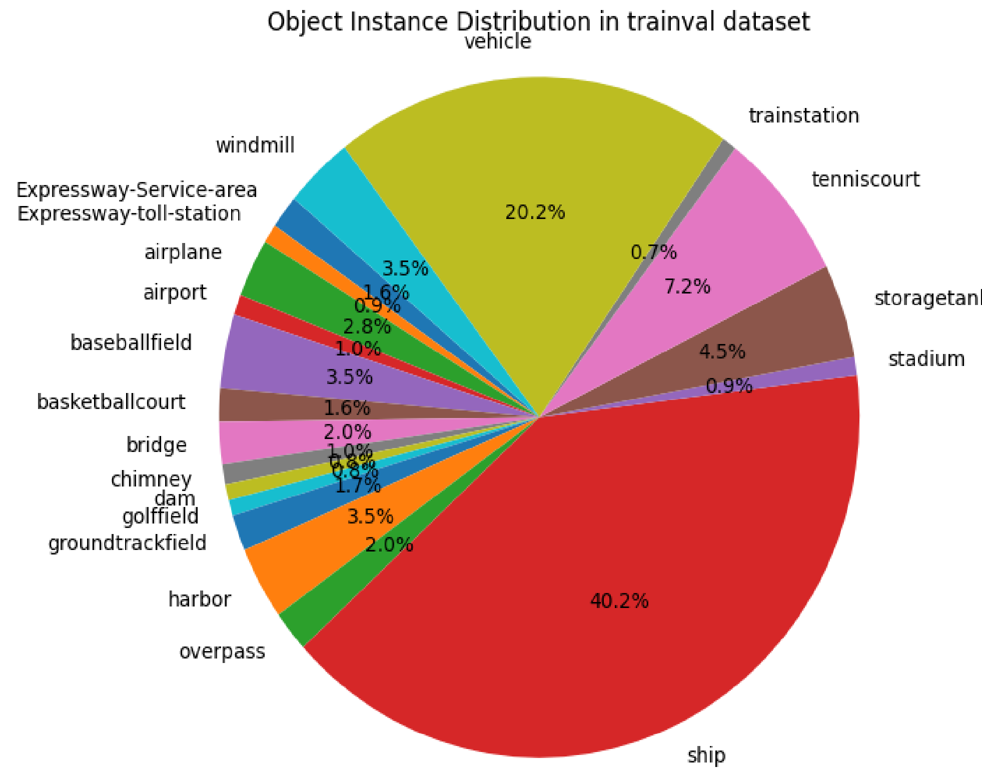
4 Parameters + Angle parameter

(X_left_top, Y_left_top, X_right_top, Y_right_top, X_left_bottom, Y_left_bottom, X_right_bottom, Y_right_bottom)

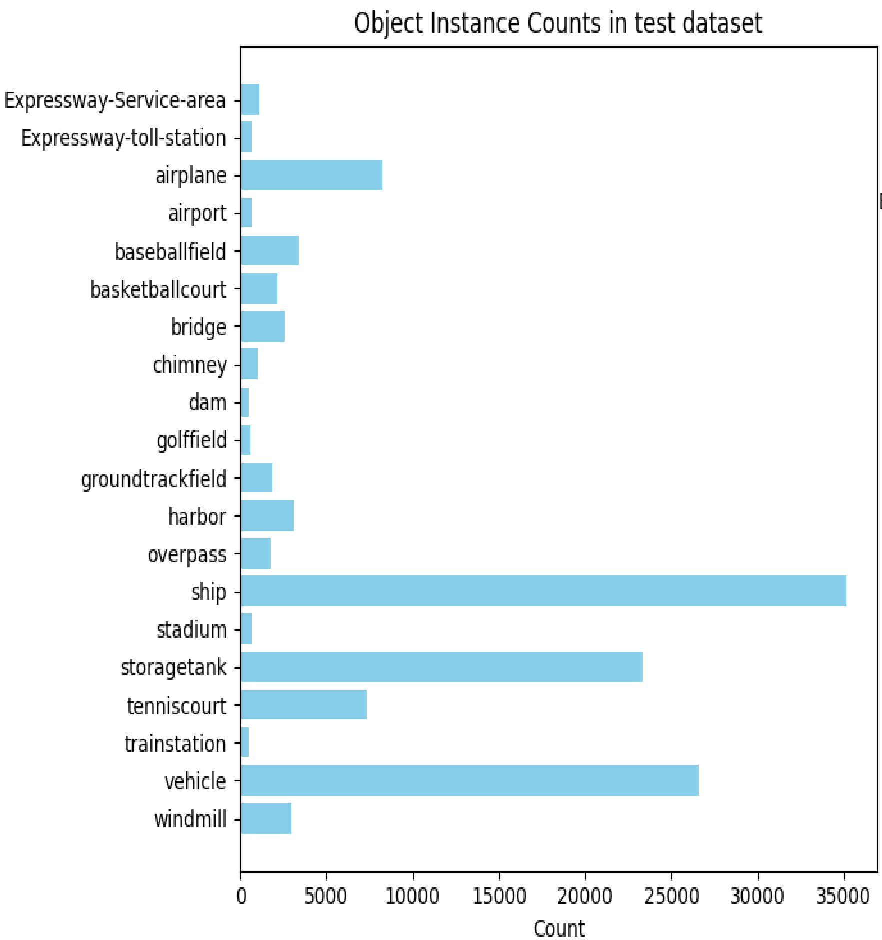
Lei Gao et. al., "**Center-Ness and Repulsion: Constraints to Improve Remote Sensing Object Detection via RepPoints**" *Remote Sens.* **2023**, *15*(6), 1479; **DOI: 10.3390/rs15061479**

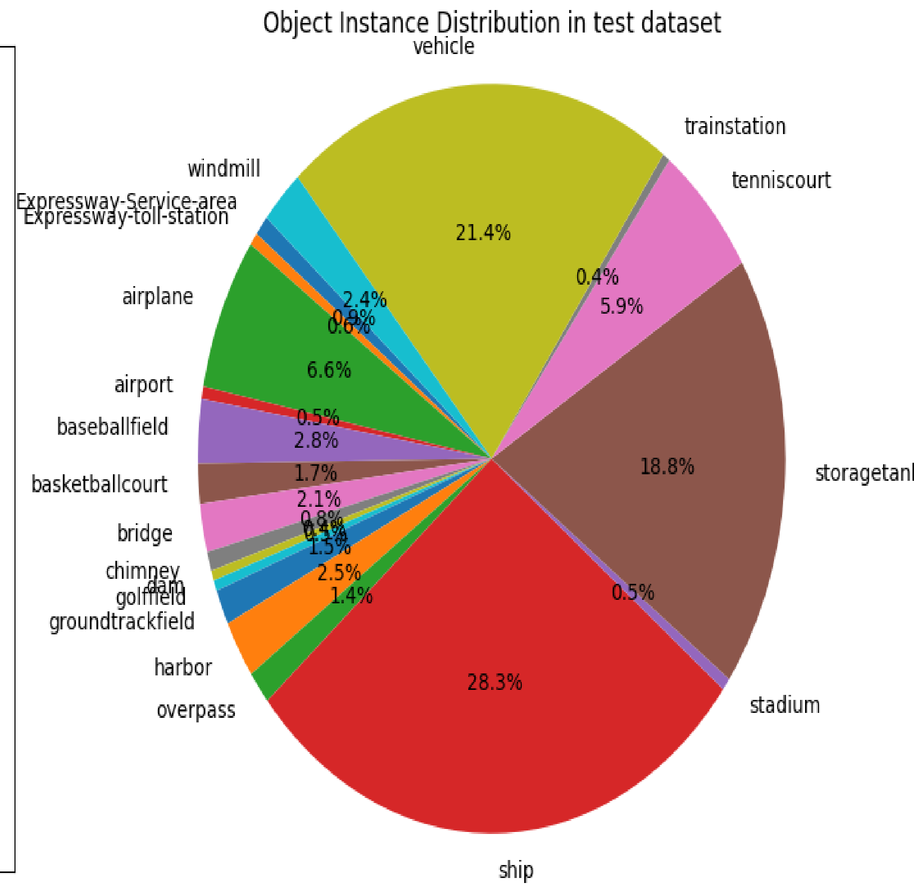
Object Instance counts in Train Val dataset





Object Instance counts in Test dataset





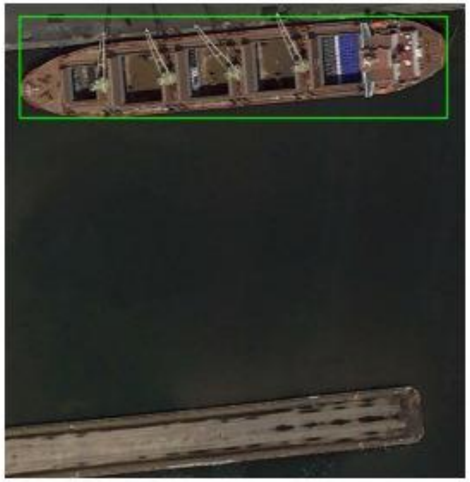
Aspect ratios

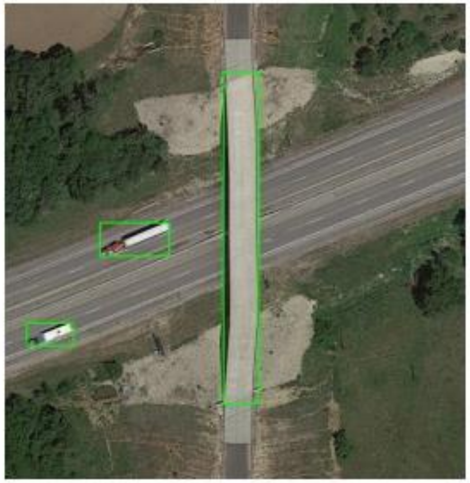
Image Aspect Ratio

- Every image is of shape 800*800 pixels, i.e. it's a square!
- Aspect ratio exactly **1**

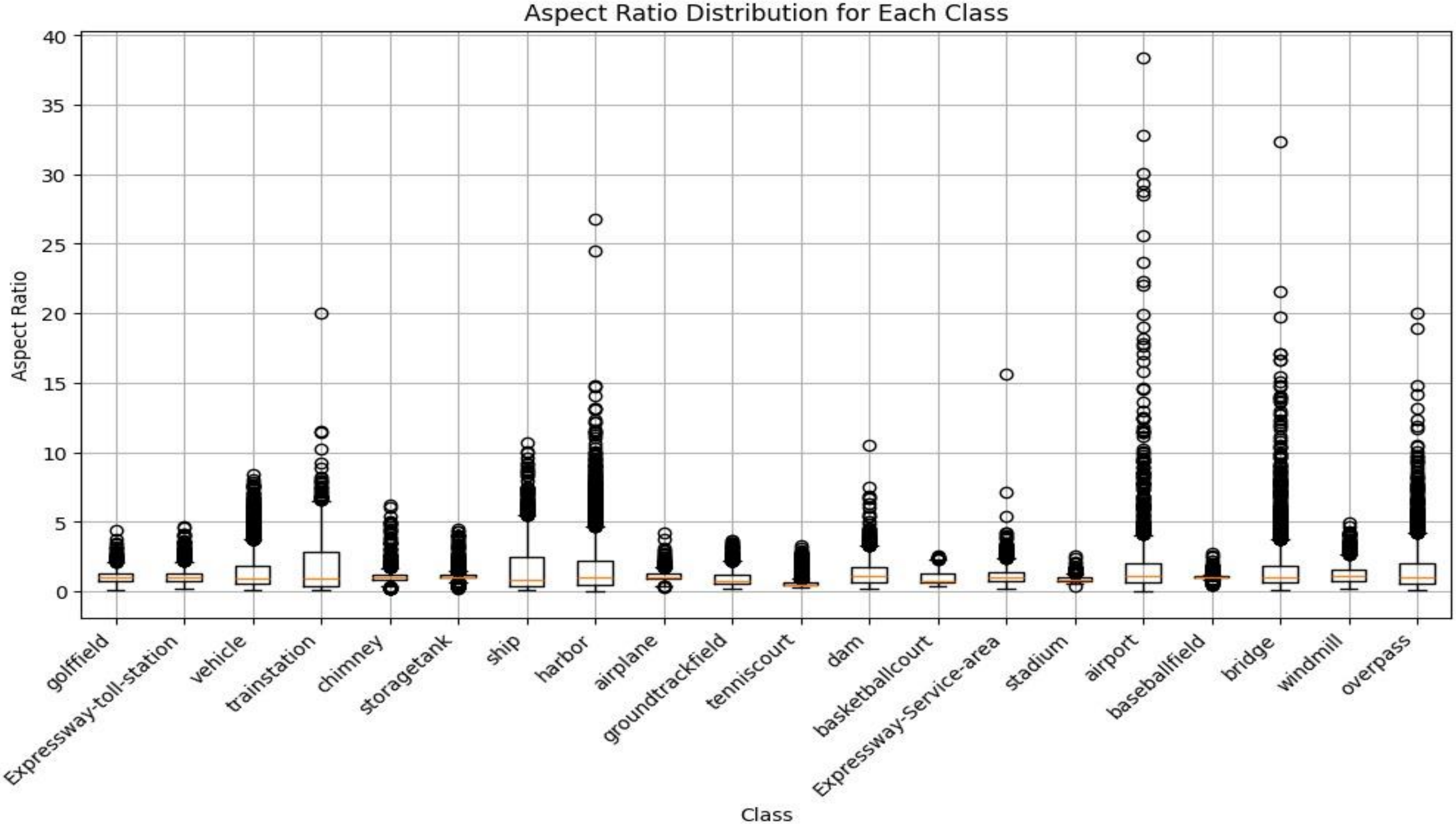
Object Aspect Ratio

- Ratio of the width to the height of a specific object within an image

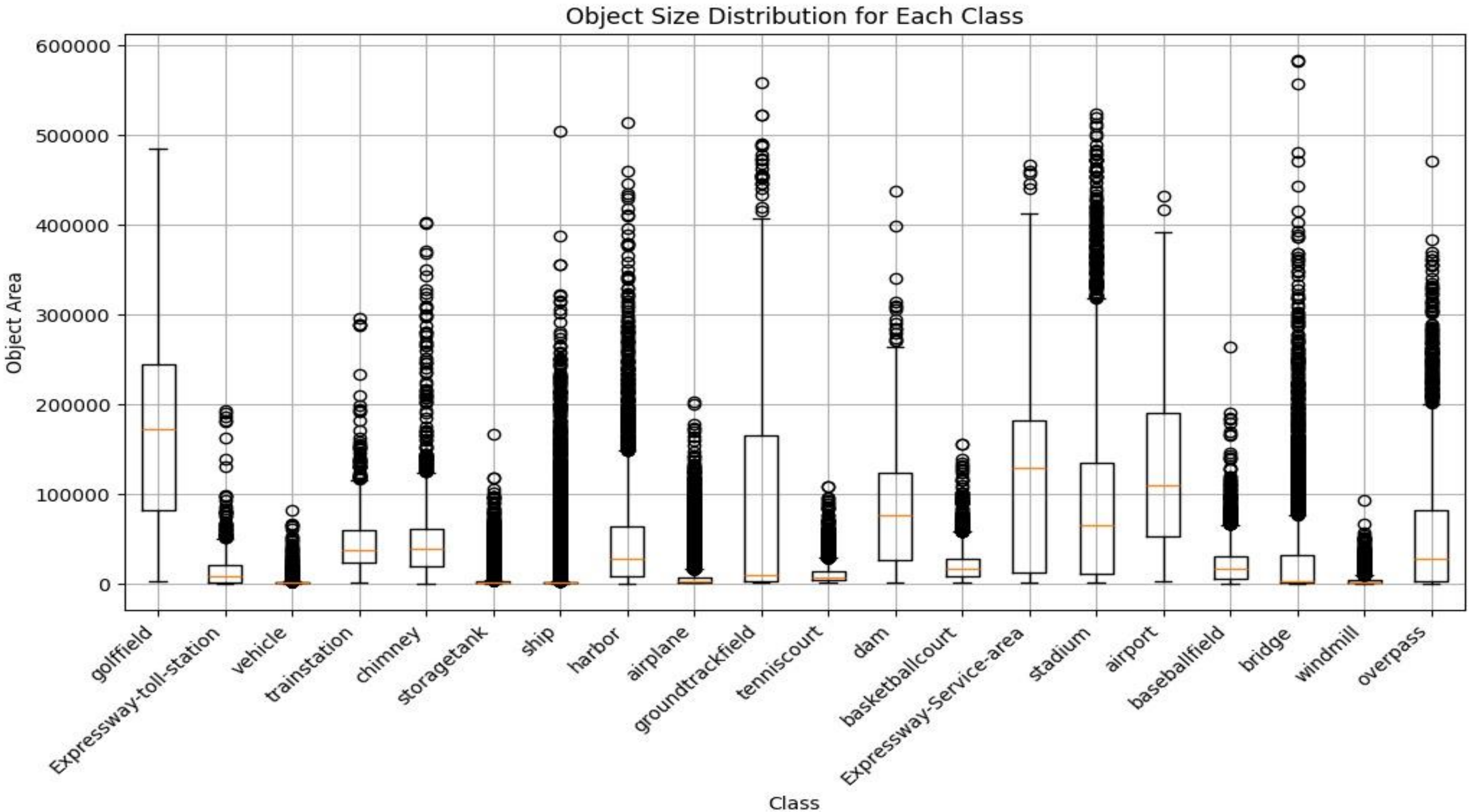




Object aspect ratio distribution



Object size distribution for each class

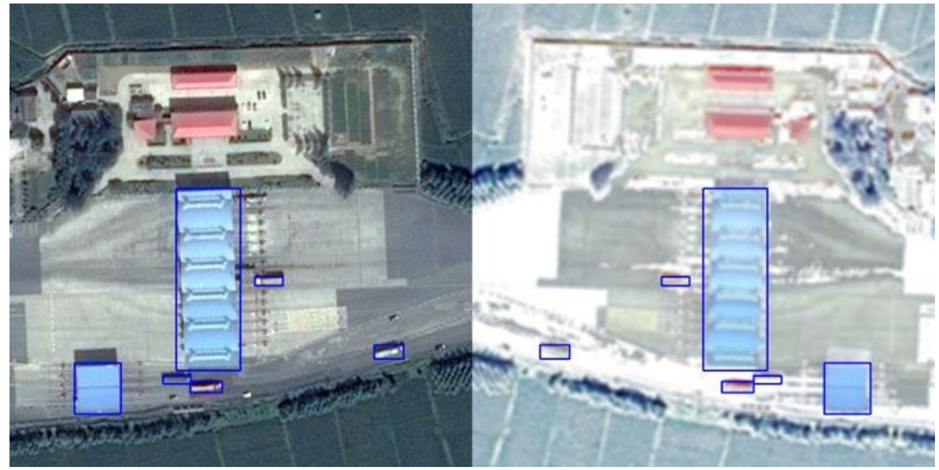


Outline

1. Introduction and motivation
2. DIOR dataset
3. **Regularization and Hyperparameters**
4. Metrics
5. Models and Experimental Setup
6. Result
7. Summary and References

Regularization and Hyperparameters

- **Transformation Types :** Horizontal Flip, RandomFog and Random Snow, & RandomBrightnessContrast



- **Optimizer** : Stoachstic Gradient Descent, Adam

- **Learning Rate** : 0.01, 0.001

https://albumentations.ai/docs/api_reference/full_reference/?query=gaussian#albumentations.augmentations

Data Distribution

Dataset Split

- Test set contains 11738
- Train Val set contains 11725

Split set	Used distribution
Training set	**80%** of Train Val set
Validation set	**20%** of Train Val set
Test set	**100%** of Test set

YOLOv8 Preprocessing

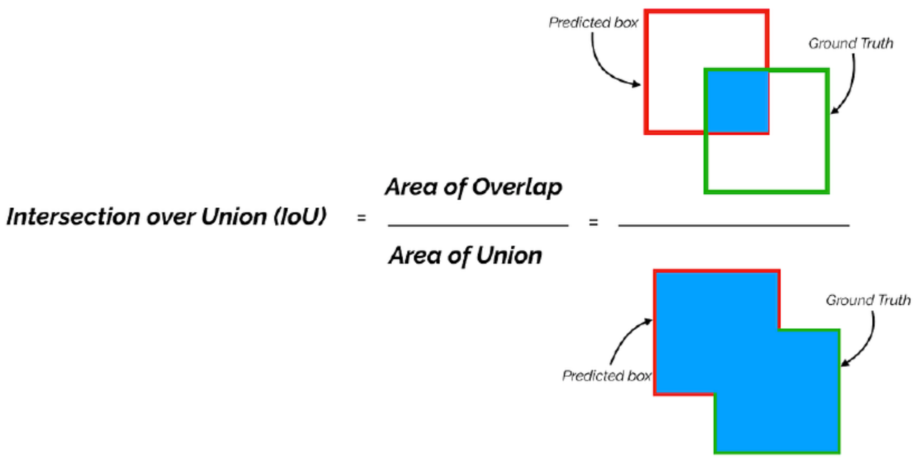
- SEED = 42
- TARGET SIZE = 640
- BATCH SIZE = 4
- PATIENCE = 20

Outline

Metrics

Intersection over Union (IoU)

- Measure the overlap between predicted bounding boxes and ground truth bounding boxes
- Based on the IoU threshold we select the following four metrics:
  - True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN)



Metrics

Precision

- quantify the **accuracy of the positive predictions** made by the model
- assesses how well the model distinguishes true objects from false positives

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Positives (FP)}}$$

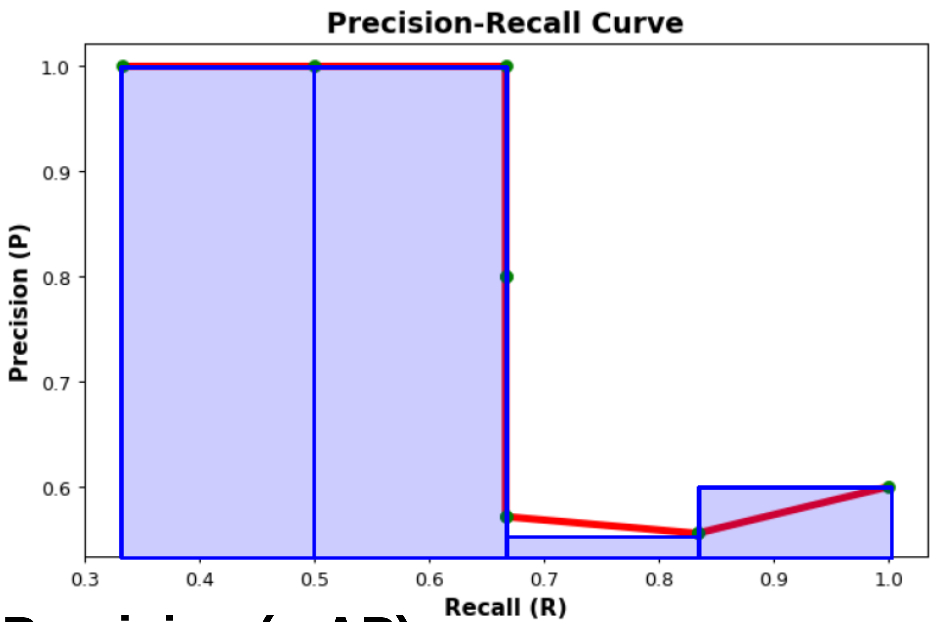
Recall

- measures the percentage of **actual positive instances** that were **correctly detected** by the model
- also known as **sensitivity** or true positive rate

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negatives (FN)}}$$

Metrics

- **Average Precision (AP) :** Area under the **precision-recall curve**

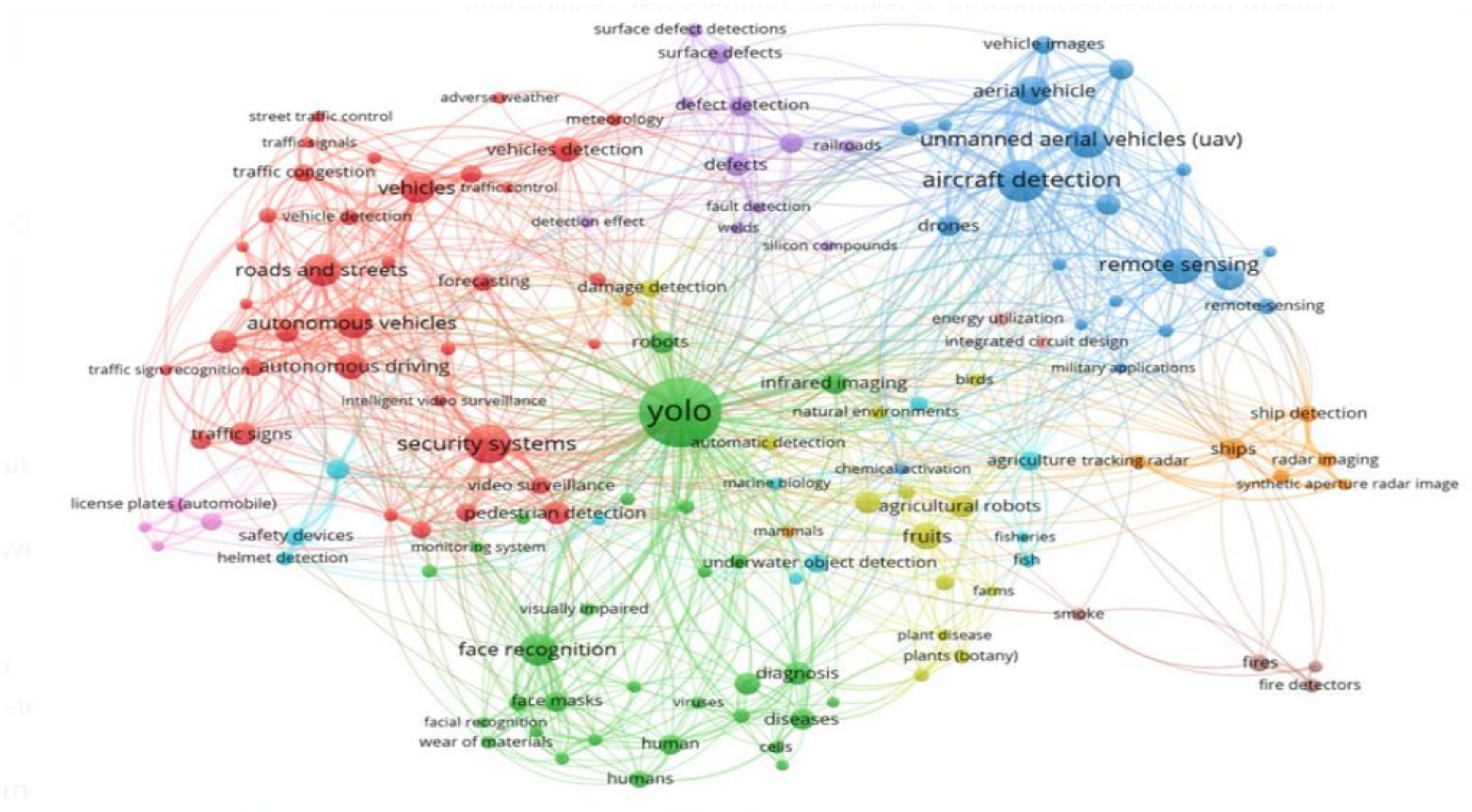


- **Mean Average Precision (mAP) :**
  - Averaging the **AP** scores across all classes
  - Single scalar value that summarize overall performance of the model
  - mAP@50 : Assesses small object well
  - mAP@50-95 may not accurately reflect the nuances of small object detection

Outline

YOLO - You Only Look Once

- **YOLOv8** : Developed by Ultralytics
- **YOLO NAS** : Developed by Deci AI

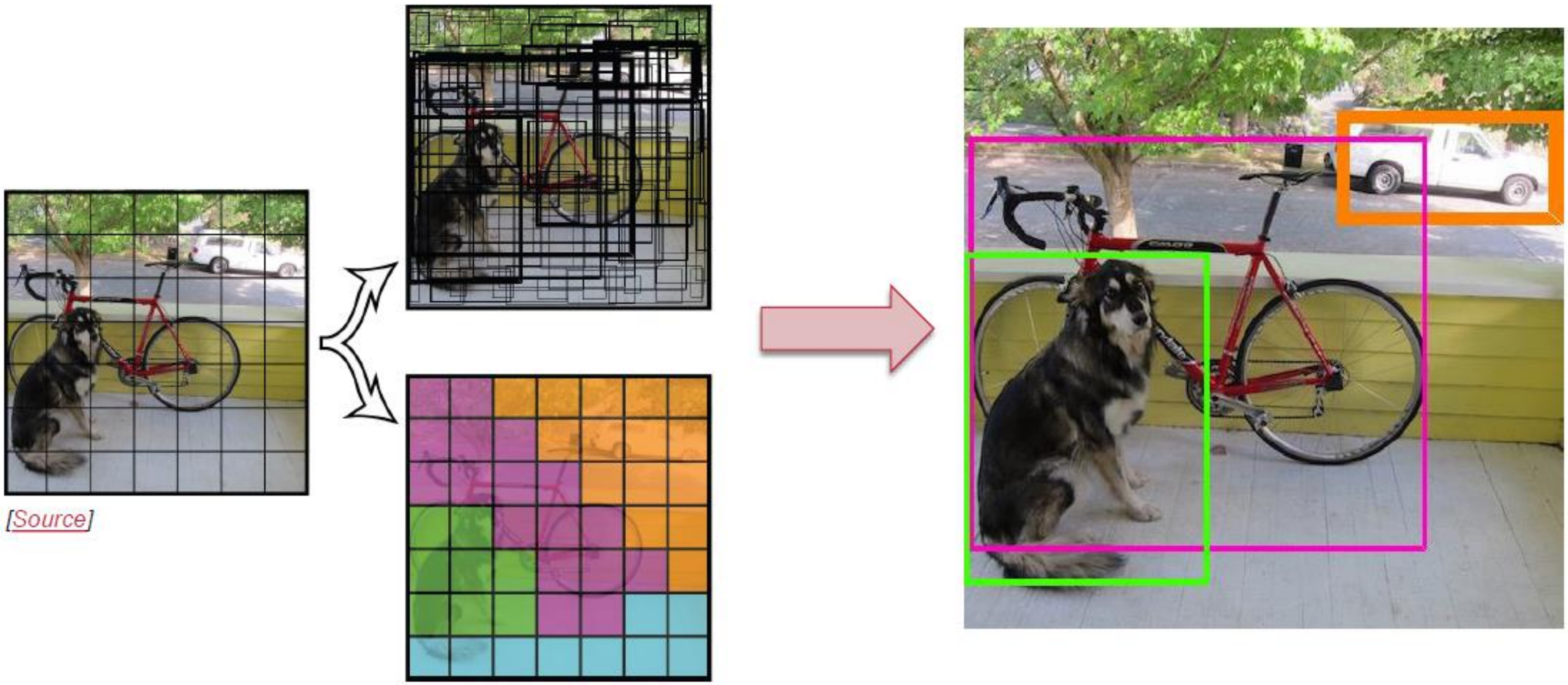


https://docs.ultralytics.com/#how-can-ultralytics-yolo-be-used-for-real-time-object-tracking

Motivation for YOLO models

- Single-stage detectors
- Real-time performance
- Removes the Region Proposal step
- Multiple models available
- Minimal latency
- Trade-off between speed and accuracy

Building blocks of the algorithm

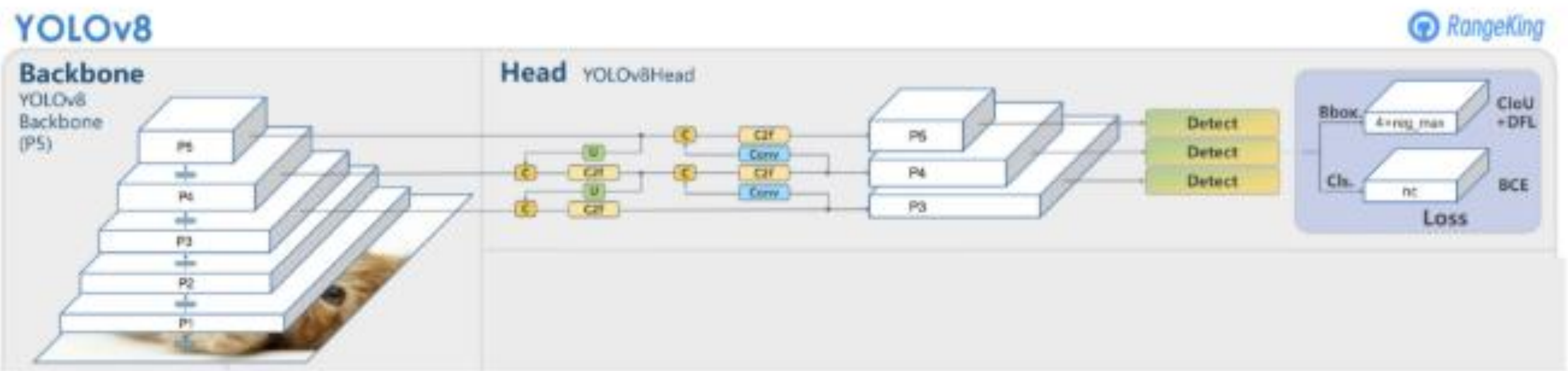


[Source]

- Residual Blocks
- Bounding box regression
- IOU
- Non-Maximum Suppression

YOLOv8

- **Backbone :** CSPDarknet53
- **Head :** multiple convolutional layers + Fully connecteed layers
- **Feature pyramid network:** allows multi-scaled object detection
- Each convolution has batch normalization and SiLU activation
- Sigmoid function used as the activation function for the objectness score, representing the probability that the bounding box contains an object.
- Softmax function for the class probabilities



Dilion Reis *et. al.*, "**Real-Time Flying Object Detection with YOLOv8**", DOI: https://doi.org/10.48550/arXiv.2305.09972

YOLOv8

- Anchor free, predicts fewer boxes and faster Non-Maximum Suppression process
- losses have improved object detection performance, particularly when dealing with smaller objects.
- Robust community
- Real time performance

Model	Size (pixels)	mAP 50-95 (val)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	3.2	8.7
YOLOv8s	640	44.9	11.2	28.6
YOLOv8m	640	50.2	25.9	78.9
YOLOv8l	640	52.9	43.7	165.2
YOLOv8x	640	53.9	68.2	257.8

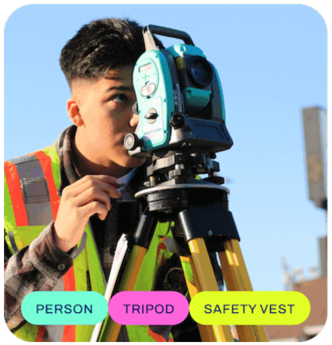
Table 2. Performance and parameter specification for the five models available for YOLOv8

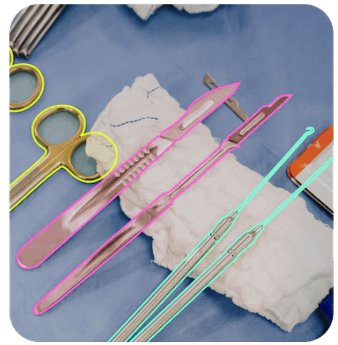
Dilion Reis *et. al.*, "**Real-Time Flying Object Detection with YOLOv8**", DOI: https://doi.org/10.48550/arXiv.2305.09972

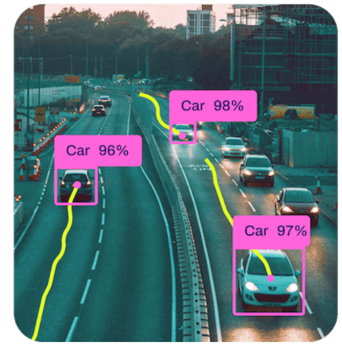
YOLOv8

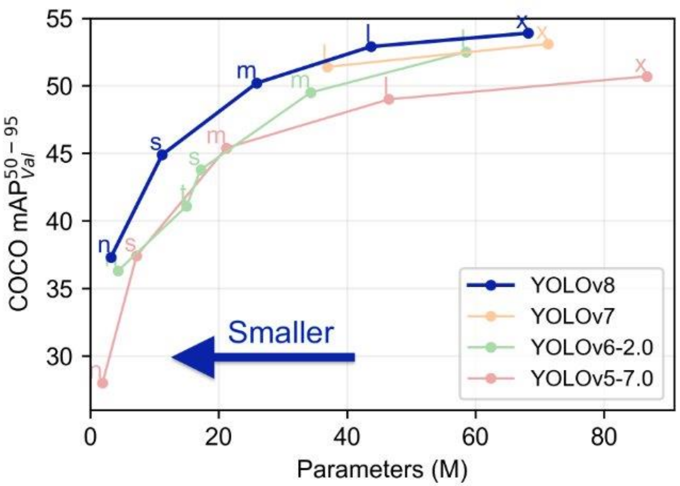
Different Models available:

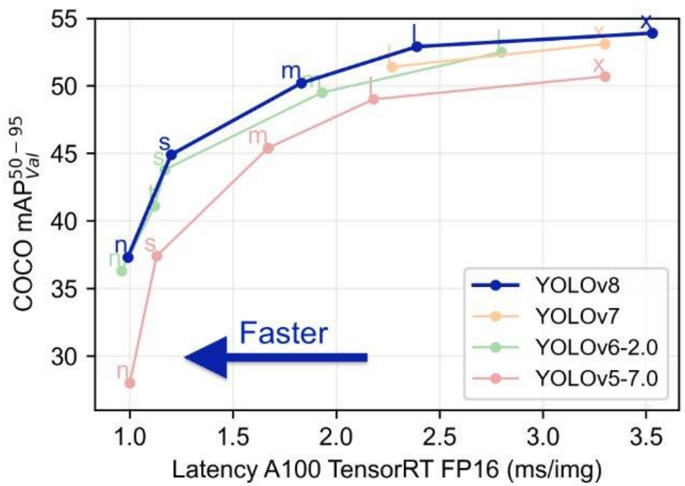
Classify | Detect | Segment | Track | Pose











Technische Universität Braunschweig

https://github.com/ultralytics/ultralytics

Dilion Reis *et. al.*, "**Real-Time Flying Object Detection with YOLOv8**", DOI: https://doi.org/10.48550/arXiv.2305.09972

igp

YOLOv8

Model Architecture

Model	YOLOv8
Nano	▪ 225 layers ▪ 3,014,748 parameters ▪ 3,014,732 gradients ▪ 8.2 GFLOPs
Medium	▪ 295 layers ▪ 25,867,900 parameters ▪ 25,867,884 gradients ▪ 79.1 GFLOPs

Initial Experimental Setup : Training

Name	YOLOv8 Models	Optimizer	Learning rate	Epochs	Precision	Recall	mAP @50	mAP @50-90
Exp 01	Sample Run							
Exp 02	Nano	Adam	0.01	145	0.74	0.676	0.714	0.489
Exp 03	Nano	SGD	0.01	200	0.882	0.792	0.855	0.655
Exp 04	Nano	Adam	0.001	200	0.87	0.781	0.837	0.635
Exp 05	Nano	SGD	0.001	200	0.856	0.755	0.806	0.586
Exp 06	Medium	Adam	0.01	144	0.688	0.643	0.654	0.452
Exp 07	**Medium**	**SGD**	**0.01**	**200**	**0.886**	**0.749**	**0.892**	**0.713**
Exp 08	Medium	Adam	0.001	200	0.872	0.792	0.854	0.662
Exp 09	Medium	SGD	0.001	200	0.894	0.804	0.858	0.653

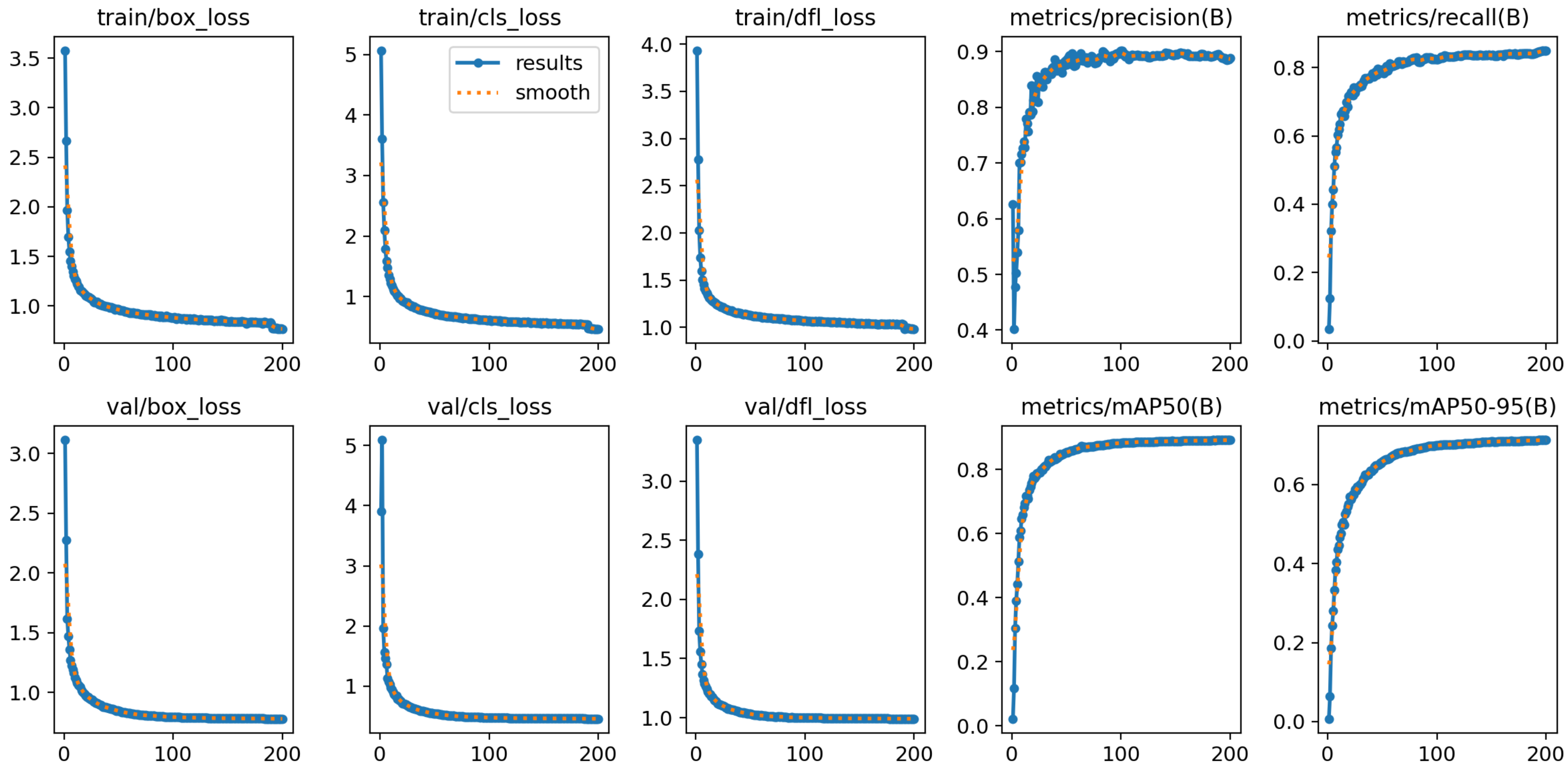
- Best models based on both mAP@50 and mAP@50-90 → Exp 07 – Exp 09 – Exp 08

Experimental 07: Training

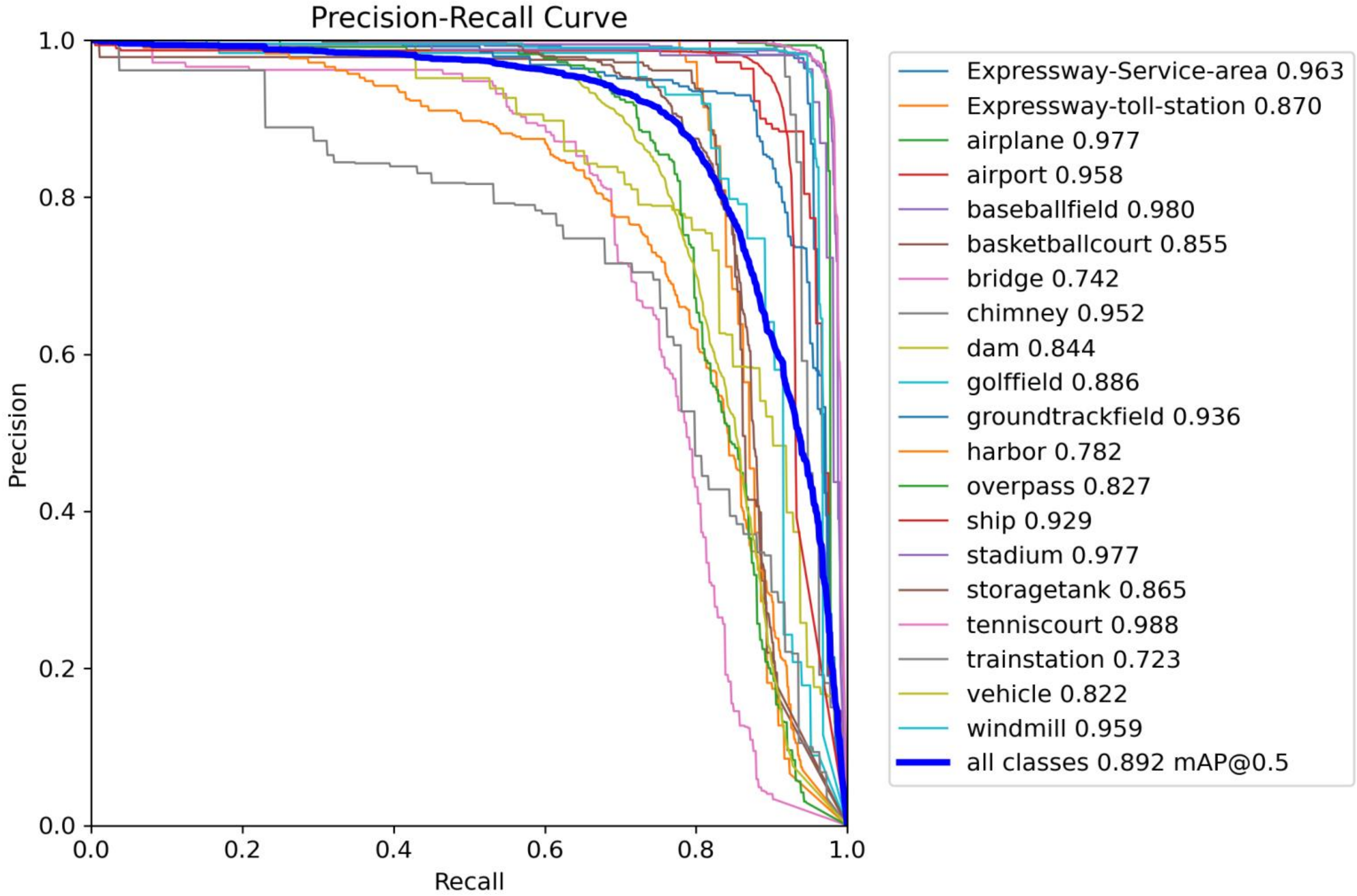
```
200 epochs completed in 23.669 hours.
Optimizer stripped from YOLOv8-Final-Project/Experiment-7-18-08/weights/last.pt, 52.1MB
Optimizer stripped from YOLOv8-Final-Project/Experiment-7-18-08/weights/best.pt, 52.1MB

Validating YOLOv8-Final-Project/Experiment-7-18-08/weights/best.pt...
Ultralytics YOLOv8.2.78 🚀 Python-3.11.9 torch-2.4.0+cu121 CUDA:0 (Tesla P100-SXM2-16GB, 16276MiB)
YOLOv8m summary (fused): 218 layers, 25,851,340 parameters, 0 gradients, 78.7 GFLOPs
                      all       2345      13015      0.886      0.849      0.892      0.713
Expressway-Service-area        117        227       0.89      0.952      0.963      0.791
Expressway-toll-station        124        131      0.935      0.809       0.87      0.713
                 airplane        143        317      0.972       0.97      0.977      0.907
                  airport        118        121      0.881      0.917      0.958      0.777
            baseballfield        231        496      0.952      0.974       0.98      0.907
           basketballcourt        154        282      0.879      0.823      0.855      0.788
                   bridge        169        273      0.854      0.641      0.742       0.52
                  chimney         78        133      0.992      0.906      0.952       0.87
                      dam        104        112      0.755      0.797      0.844      0.513
                golffield         73         83      0.817      0.843      0.886      0.738
         groundtrackfield        193        243      0.831      0.905      0.936      0.808
                   harbor        128        440      0.741      0.736      0.782      0.576
                 overpass        193        276      0.853      0.757      0.827      0.614
                     ship        256       4870      0.947      0.902      0.929      0.645
                  stadium        108        109      0.931      0.954      0.977      0.874
              storagetank        156        625      0.927      0.757      0.865       0.74
              tenniscourt        249       1015      0.973      0.963      0.988       0.94
             trainstation        105        109      0.701      0.732      0.723      0.431
                  vehicle        654       2754      0.912      0.691      0.822      0.538
                 windmill        158        399      0.982      0.946      0.959      0.567
Speed: 0.2ms preprocess, 9.8ms inference, 0.0ms loss, 1.3ms postprocess per image
Results saved to ESC[1mYOLOv8-Final-Project/Experiment-7-18-08ESC[0m
```

Experimental 07: Training metrics



Experimental 07: Training metrics



Initial Experimental Setup : Testing

Name	YOLOv8 Models	Optimizer	Learning rate	Epochs	Precision	Recall	mAP @50	mAP @50-90
Exp 01	Sample Run							
Exp 02	Nano	Adam	0.01	145	0.687	0.52	0.564	0.366
Exp 03	Nano	SGD	0.01	200	0.832	0.661	0.739	0.523
Exp 04	Nano	Adam	0.001	200	0.825	0.637	0.723	0.508
Exp 05	Nano	SGD	0.001	200	0.808	0.591	0.666	0.446
Exp 06	Medium	Adam	0.01	144	0.636	0.511	0.528	0.349
Exp 07	Medium	SGD	0.01	200	0.851	0.732	0.806	0.594
Exp 08	Medium	Adam	0.001	200	0.82	0.675	0.751	0.54
Exp 09	Medium	SGD	0.001	200	0.833	0.671	0.746	0.523

Best models based on both mAP@50 and mAP@50-90 → Exp 07 – Exp 09 – Exp 08 – Exp 03

Experimental 07: Inference

```
Validating Experiment-7-18-08...
Ultralytics YOLOv8.2.78 🚀 Python-3.11.9 torch-2.4.0+cu121 CUDA:0 (Tesla P100-SXM2-16GB, 16276MiB)
YOLOv8m summary (fused): 218 layers, 25,851,340 parameters, 0 gradients, 78.7 GFLOPs
                    all      11738     124445      0.851      0.732      0.806      0.594
Expressway-Service-area        565       1085      0.775      0.923      0.923      0.685
Expressway-toll-station        634        688      0.877      0.715      0.765      0.594
               airplane        705       8212      0.962      0.739      0.869      0.625
                airport        657        666      0.843      0.868      0.909      0.684
          baseballfield       1311       3434      0.951      0.733      0.874      0.732
         basketballcourt        705       2146      0.932      0.888      0.917      0.827
                 bridge       1299       2589      0.765      0.418      0.545      0.316
                chimney        447       1031      0.963      0.743      0.804      0.696
                    dam        503        538      0.713      0.735      0.763      0.471
              golffield        491        575      0.806      0.814      0.854      0.693
        groundtrackfield       1324       1885      0.732      0.864      0.855      0.685
                 harbor        813       3105      0.774      0.649        0.7      0.517
               overpass       1101       1782      0.786      0.591      0.674      0.459
                   ship       1404      35186      0.942      0.871      0.929      0.625
                stadium        618        672      0.856      0.654      0.823      0.648
            storagetank        894      23361       0.97      0.594      0.814      0.545
            tenniscourt       1346       7343      0.949      0.899      0.935      0.818
           trainstation        501        509      0.595      0.721      0.686       0.42
                vehicle       3308      26640      0.897      0.398      0.576      0.343
               windmill        809       2998      0.942      0.829      0.907      0.489
Speed: 0.4ms preprocess, 28.5ms inference, 0.0ms loss, 2.4ms postprocess per image
Results saved to ESC[1mruns/detect/val8ESC[0m
```

Under-performing classes

Experiment Tuning Setup : Training

- Augmented images used in training dataset
- Classes are selected based on :
    1. Underperforming classes in **Experiment 07**
    2. Number of objects instances in training dataset
- Selected classes are Bridge, Vehicle, Trainstation, Overpass, Dam

Before Augmentation

Name	YOLOv8 Models	Optimizer	Learning rate	Epochs	Precision	Recall	mAP @50	mAP @50-90
Exp 06	Medium	Adam	0.01	144	0.688	0.643	0.654	0.452
Exp 07	Medium	SGD	0.01	200	0.886	0.749	0.892	0.713

After Augmentation

Name	YOLOv8 Models	Optimizer	Learning rate	Epochs	Precision	Recall	mAP @50	mAP @50-90
Exp 06	Medium	Adam	0.01	144	0.643	0.607	0.612	0.408
Exp 07	Medium	SGD	0.01	200	0.894	0.842	0.886	0.698

Experiment Tuning Setup : Testing

Before Augmentation

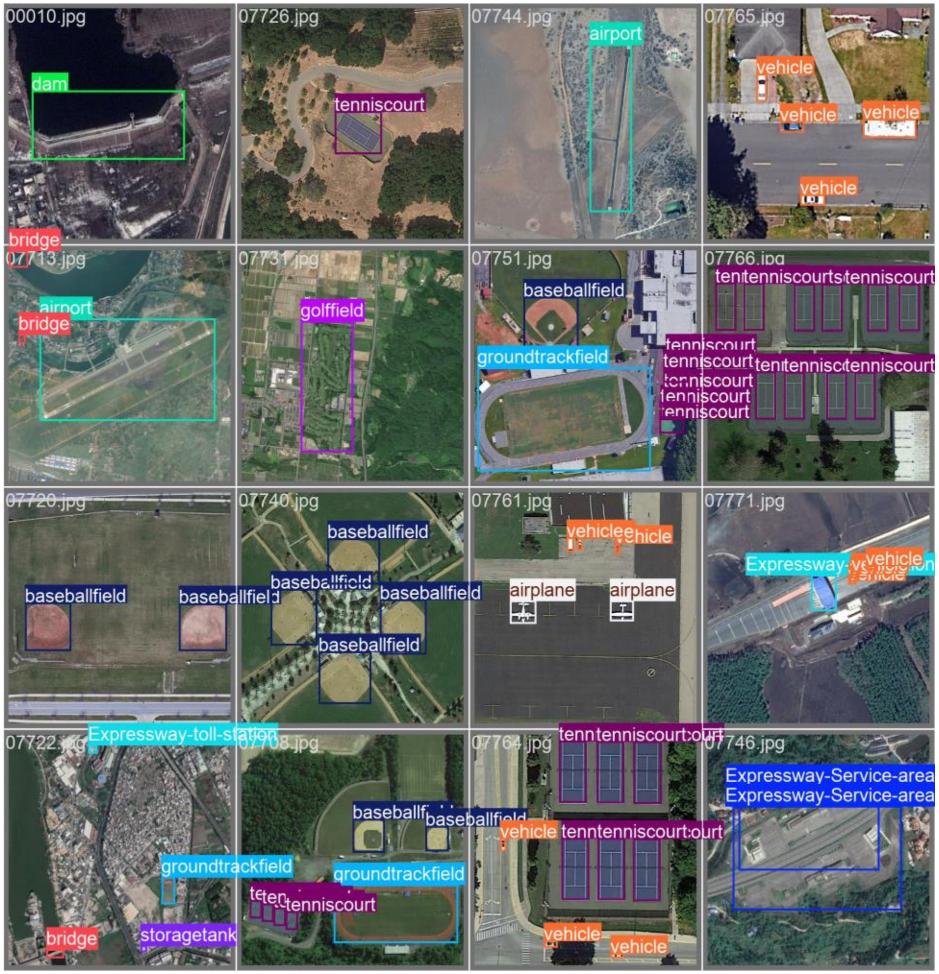
Name	YOLOv8 Models	Optimizer	Learnin g rate	Epochs	Precisi on	Recall	mAP @50	mAP @50-90
Exp 06	Medium	Adam	0.01	144	0.636	0.511	0.528	0.349
Exp 07	Medium	SGD	0.01	200	0.851	0.732	0.806	0.594

After Augmentation

Name	YOLOv8 Models	Optimize r	Learning rate	Epochs	Precision	Recall	mAP @50	mAP @50-90
Exp 06	Medium	Adam	0.01	144	0.608	0.465	0.485	0.309
Exp 07	Medium	SGD	0.01	200	0.853	0.728	0.795	0.577

YOLOv8 Results

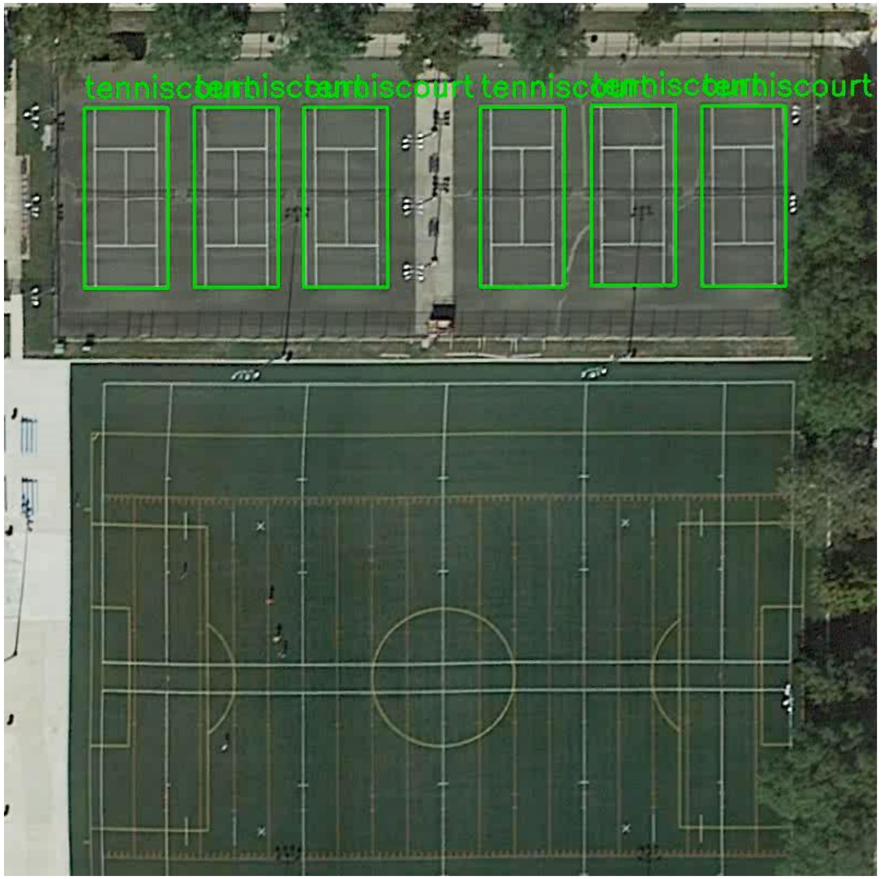
Ground Truth Labels



Predictions YOLOv8m model (Exp 07)

YOLOv8 Results

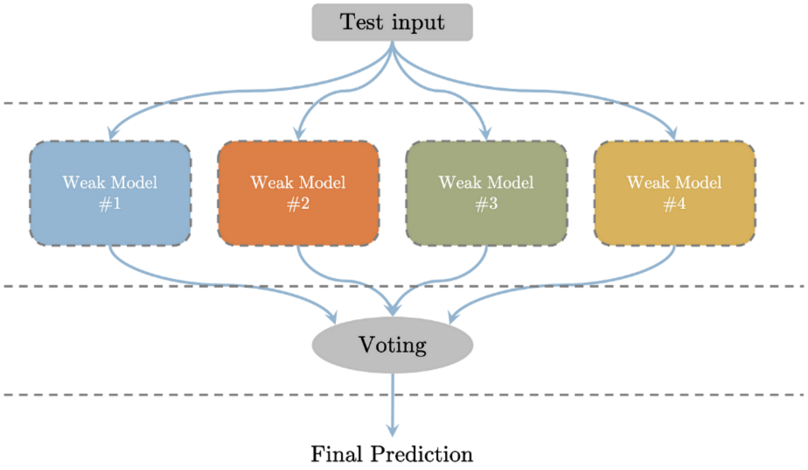
Ground Truth Labels



Predictions YOLOv8m model (Exp 07)

Ensemble Methods

- Combine decisions from multiple models to improve the overall performance
- Combining predictions from two independently trained YOLOv8 models using concatenation followed by Non-Maximum Suppression (NMS)
- Increase diversity of prediction
- **Ensemble 1 :** Experiment 7 + Experiment 3
- **Ensemble 2 :** Experiment 7 + Experiment 9



Ensemble Methods

Model	Precision	Recall	mAP@0.50
Ensemble 1	0.6949	0.733	0.48
Ensemble 2	0.6944	0.7362	0.4772

IoU Threshold : 0.4

Model	Precision	Recall	mAP@0.50
Ensemble 1	0.6961	0.7348	0.4839
Ensemble 2	0.6956	0.7382	0.4822

IoU Threshold : 0.5

Model	Precision	Recall	mAP@0.50
Ensemble 1	0.6965	0.7359	0.4860
Ensemble 2	0.6958	0.7393	0.4847

IoU Threshold : 0.6

Wandb Links

1. Experiment 02
2. Experiment 03
3. Experiment 04
4. Experiment 05
5. Experiment 06
6. Experiment 06 (Tuning)
7. Experiment 07
8. Experiment 07 (Tuning)
9. Experiment 08
10. Experiment 09

Yolo-NAS: Model Discovery

- Released in May 2023 by Deci
- Network Architecture Search (NAS)
  - Sub-field within automated ML
  - Development of ML models that automatically designs and configures deep neural network architectures
  - Efficient discovery of high-performing architectures considering various factors
  - Other examples: MobileNetV3, EfficientDet (designed by humans and optimised using NAS)
  - Required extensive resourced and limited to smaller number of organisations
  - AutoNAC –Proprietory NAS Technology of Deci

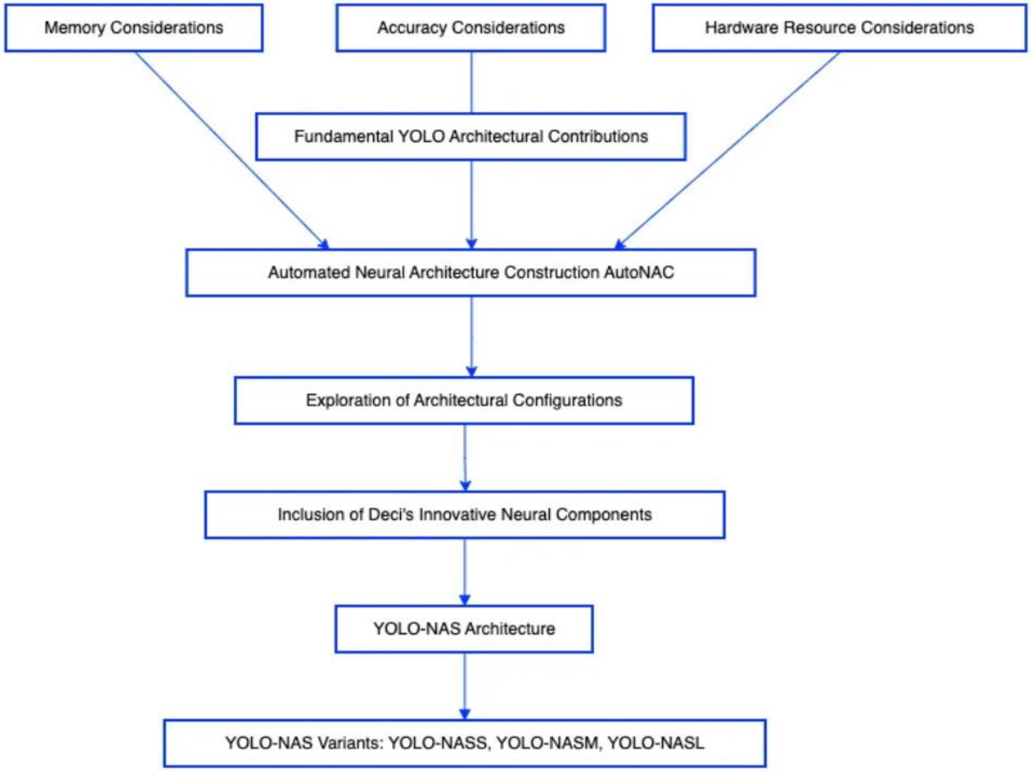
Yolo-NAS: Model Characteristics

- Facilitates discovery of architectures considering hardware availability, performance targets, quantisation etc.

- Quantisation- Making the neural network ‘smaller’

- Reduction of bit-representation of numeral values

- Hybrid quantisation method to combat info loss to run on edge devices

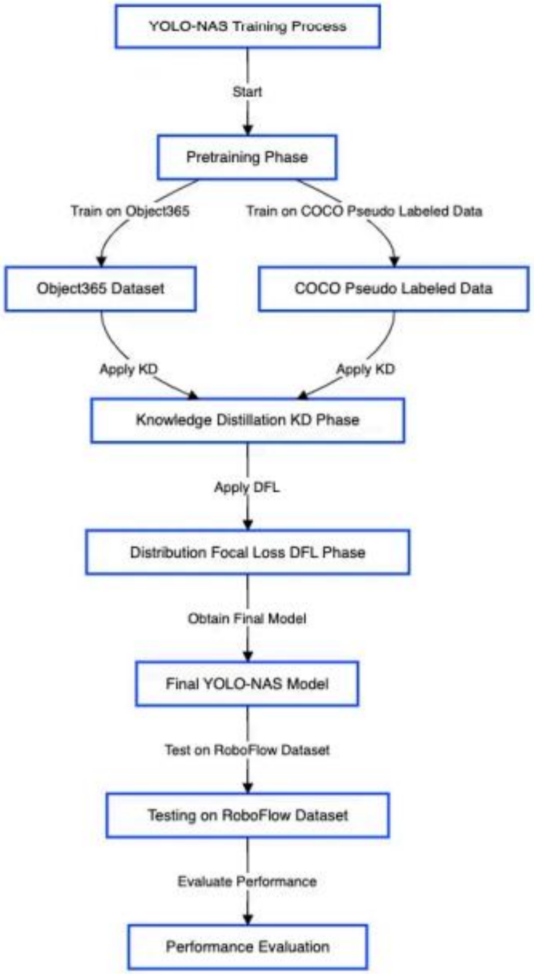


Process leading to discovery of Yolo-NAS

Ref: https://richmondalake.medium.com/yolo-nas-uncovered-essential-insights-and-implementation-techniques-for-machine-learning-engineers-87ee266b37f6

Yolo-NAS: Model Characteristics

- Incorporates attention mechanism to focus on parts of image containing the target object
- Trained initially on Object 365 (2 M images across 365 categories
- Coco pseudo-labelled dataset (semi-supervised technique) adds 125k images
- Training incorporates Knowledge Distribution and Distribution Focal Loss
  - KD: to reduce computational resource by training a simpler student model to perform as the original model
  - DFL: Modification of focal loss function. Enables model to navigate variability in target object sizes



Yolo-NAS: Model Performance

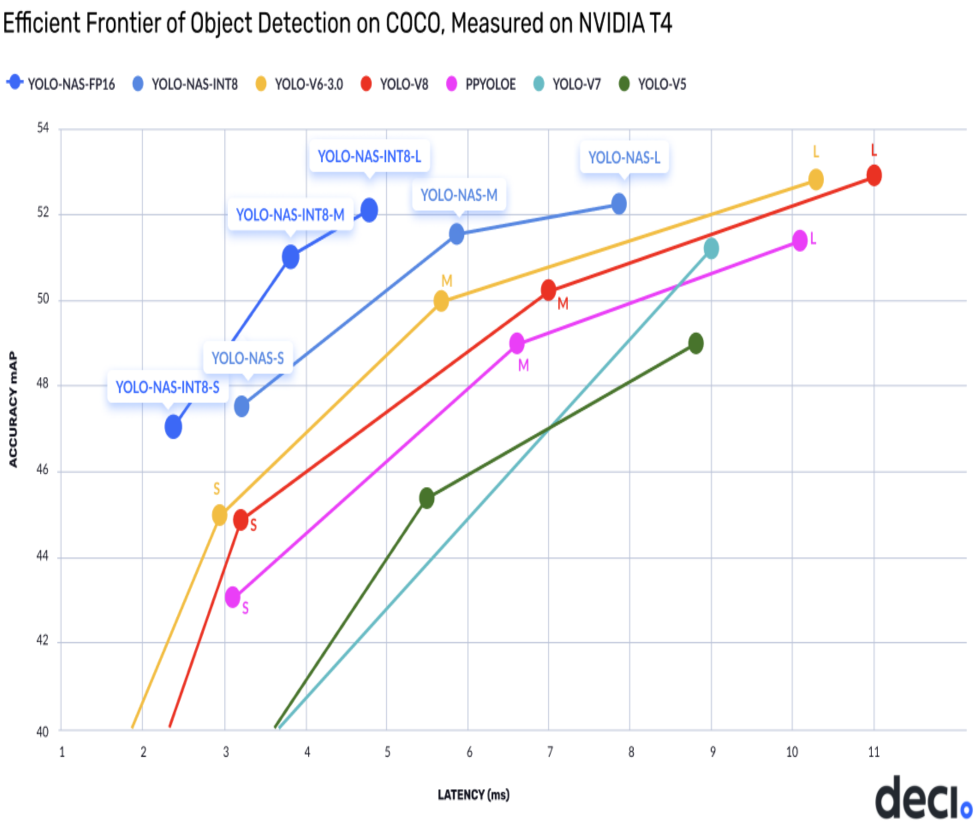


Fig-1

Fig-2: Coco 2017 Dataset for 640x640 images on Nvidia T4 GPU

Model	mAP	Latency (ms)
YOLO-NAS S	47.5	3.21
YOLO-NAS M	51.55	5.85
YOLO-NAS L	52.22	7.87
YOLO-NAS S INT-8	47.03	2.36
YOLO-NAS M INT-8	51.0	3.78
YOLO-NAS L INT-8	52.1	4.78

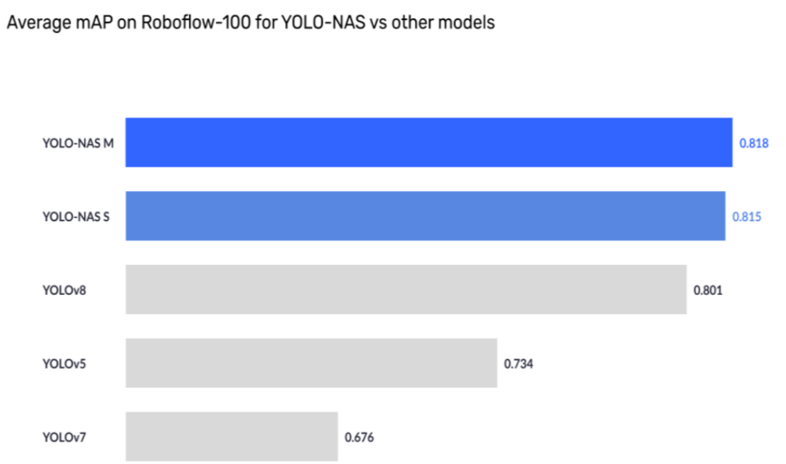


Fig-3

Yolo-NAS

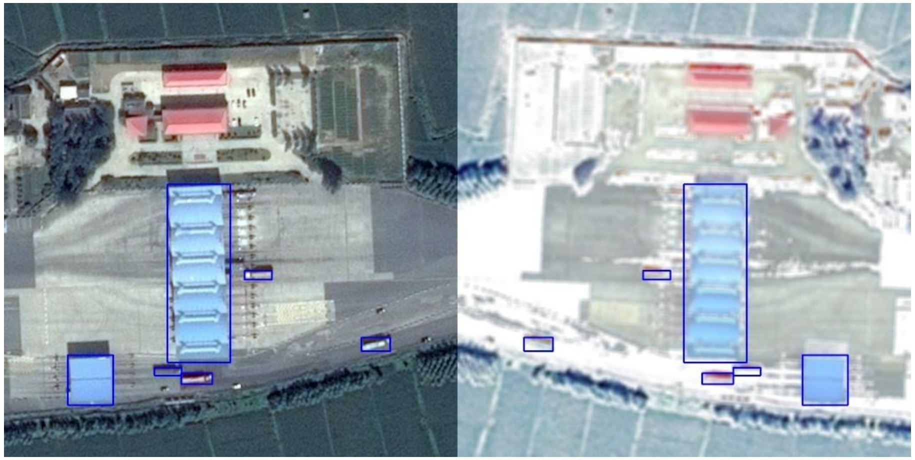
YOLO-NAS is a state-of-the-art object detection model leveraging advanced Neural Architecture Search (NAS) technology. It addresses the limitations of previous YOLO models by introducing features like quantization-friendly basic blocks and sophisticated training schemes

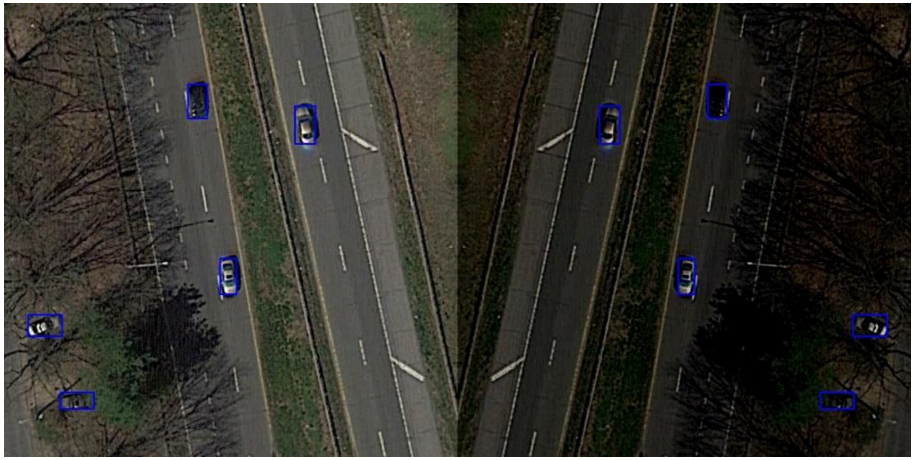
Yolo-NAS: Experiment Set-Up

- Pre-trained weights available on SuperGradients, Deci’s library
- Fine-tuning possible with Trainer from SuperGradients library
- ***Dataloader***: Super-Gradeient provides ready-to-use dataloaders for some datasets
  - Coco-detection-yolo-format dataloader used for our project
  - Annotations converted from PASCAL-VOC to Yolo text format
- ***Data Augmentation***
  - Augmentation applied for under-represented classes as compared to test dataset
  - Under-represented classes : storagetank, vehicle, airplane
  - Augmentation from albumentations library
  - Pipeline includes:
    - Horizontal flip (p=0.5)
    - RandomBrightnessContrast (p=0.2)
    - RandomSnow (p=0.2)
    - RandomFog (p=0.2)
  - 3341 images are transormed with augmentations
  - Augmented images and corresponding labels are added to form full dataset

Yolo-NAS: Experimental Set-Up

Augmentation Examples





Yolo-Nas: Experimental Set-Up

- The training parameters are defined as below
  - max_epochs: 50
  - Loss: PPYoloELoss
  - Optimizer: adam and sgd
  - Metric_to_watch: mAP@0.50:0.95
  - Lr_mode: cosine
- Yolo_nas_s, yolo_nas_m and yolo_nas_l models are trained with different optimizer to evaluate the performance
- The metrics are logged during training on Weights and Biases
- The best weights are stored under the checkpoints folder
- The models are evaluated on the test set

Yolo-NAS: Model Training

- WandB Link YoloNAS-S Adam
- WandB Link YoloNAS-S SGD
- WandB Link YoloNAS-M Adam
- WandB Link YoloNAS-M SGD
- WandB Link YoloNAS-L Adam
- WandB Link YoloNAS-L SGD

Yolo-NAS: Run Summary

Model	mAP	F1	Precision	Recall
Yolo-NAS S Adam	0.45864	0.05346	0.02857	0.62784
Yolo-NAS S SGD	0.49610	0.07475	0.04050	0.64904
Yolo-NAS M Adam	0.45846	0.05637	0.03031	0.62941
Yolo-NAS M SGD	0.54484	0.08600	0.04662	0.68706
Yolo-NAS L Adam	0.46853	0.05881	0.03166	0.63515
Yolo-NAS L SGD	0.54588	0.09203	0.05013	0.69102

Yolo-NAS: Model Performance on Test Data

Model	mAP	F1	Precision	Recall
Yolo-NAS S Adam	0.42318	0.06944	0.03884	0.56913
Yolo-NAS S SGD	0.47623	0.08584	0.04825	0.60808
Yolo-NAS M Adam	0.42106	0.06771	0.03775	0.56921
Yolo-NAS M SGD	0.52334	0.09432	0.05262	0.64669
Yolo-NAS L Adam	0.43365	0.06817	0.03791	0.58749
Yolo-NAS L SGD	0.53234	0.10027	0.05641	0.65682

Yolo-NAS: Model Performance on Test Data

AP	Yolo-NAS S Adam	Yolo-NAS S SGD	Yolo-NAS M Adam	Yolo-NAS M SGD	Yolo-NAS L Adam	Yolo-NAS L SGD
Expressway-Service-area	0.417227	0.492327	0.398813	0.617081	0.439448	**0.623077**
Expressway-toll-station	0.426014	0.450274	0.409941	0.484885	0.440630	**0.494798**
airplane	0.553419	0.564674	0.547072	0.593528	0.535968	**0.605083**
airport	0.375940	0.478010	0.386172	0.564944	0.406329	**0.627903**
baseballfield	0.594900	0.624306	0.567815	0.649218	0.599659	**0.665462**
basketballcourt	0.741909	0.768957	0.762294	0.800260	0.762500	**0.813650**
bridge	0.162630	0.206824	0.150171	0.250439	0.166314	**0.253491**
chimney	0.538211	0.612456	0.540507	0.630528	0.565566	**0.647489**
dam	0.121194	0.213120	0.102045	0.265096	0.093535	**0.282639**
golffield	0.403169	0.541136	0.401836	0.606970	0.413214	**0.643129**

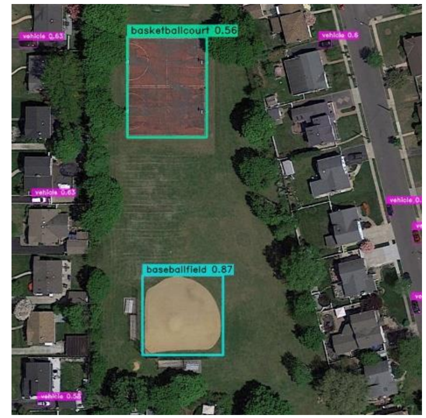
Yolo-NAS: Model Performance on Test Data

AP	Yolo-NAS S Adam	Yolo-NAS S SGD	Yolo-NAS M Adam	Yolo-NAS M SGD	Yolo-NAS L Adam	Yolo-NAS L SGD
groundtrackfield	0.562088	0.610759	0.572893	0.651818	0.569423	**0.658720**
harbor	0.349312	0.392289	0.346059	0.455936	0.352014	**0.459930**
overpass	0.336907	0.382789	0.321252	0.421299	0.344117	**0.418365**
ship	0.486952	0.526754	0.486481	0.554439	0.495917	**0.558323**
stadium	0.451598	0.515827	0.449703	**0.608344**	0.412368	0.569447
storagetank	0.446919	0.487192	0.444605	0.489653	0.469745	**0.506412**
tenniscourt	0.759473	0.771436	0.765198	0.779939	0.776781	**0.793169**
trainstation	0.114851	0.230490	0.115956	**0.311693**	0.169748	0.293088
vehicle	0.253360	0.267618	0.258598	0.305237	0.273087	**0.313810**
windmill	0.367577	0.387416	0.393888	**0.425527**	0.386726	0.418841

Yolo-NAS: Visualisation

Ground Truth Labels

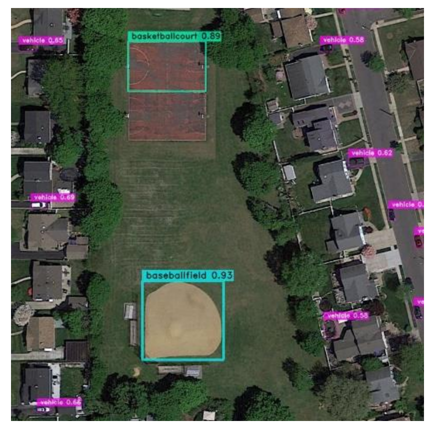
Yolo-NAS-S with Adam Opt



Yolo-NAS-M with Adam Opt

Yolo-NAS-L with Adam Opt

Yolo-NAS-S with SGD Opt



Yolo-NAS-M with SGD Opt

Yolo-NAS-L with SGD Opt

Outline

Results from DIOR paper

Table 3

Detection average precision (%) of 12 representative methods on the proposed DIOR test set. The entries with the best APs for each object category are bold-faced.

c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
Airplane	Airport	Baseball field	Basketball court	Bridge	Chimney	Dam	Expressway service area	Expressway toll station	Golf course
c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
Ground track field	Harbor	Overpass	Ship	Stadium	Storage tank	Tennis court	Train station	Vehicle	Wind mill

	Backbone	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	mAP
R-CNN	VGG16	35.6	43.0	53.8	62.3	15.6	53.7	33.7	50.2	33.5	50.1	49.3	39.5	30.9	9.1	60.8	18.0	54.0	36.1	9.1	16.4	37.7
RICNN	VGG16	39.1	61.0	60.1	66.3	25.3	63.3	41.1	51.7	36.6	55.9	58.9	43.5	39.0	9.1	61.1	19.1	63.5	46.1	11.4	31.5	44.2
RICAOD	VGG16	42.2	69.7	62.0	79.0	27.7	68.9	50.1	60.5	49.3	64.4	65.3	42.3	46.8	11.7	53.5	24.5	70.3	53.3	20.4	56.2	50.9
RIFD-CNN	VGG16	56.6	53.2	**79.9**	69.0	29.0	71.5	63.1	69.0	56.0	68.9	62.4	**51.2**	51.1	31.7	**73.6**	41.5	79.5	40.1	28.5	46.9	56.1
Faster R-CNN	VGG16	53.6	49.3	78.8	66.2	28.0	70.9	62.3	69.0	55.2	68.0	56.9	50.2	50.1	27.7	73.0	39.8	75.2	38.6	23.6	45.4	54.1
SSD	VGG16	59.5	72.7	72.4	75.7	29.7	65.8	56.6	63.5	53.1	65.3	68.6	49.4	48.1	59.2	61.0	46.6	76.3	55.1	27.4	65.7	58.6
YOLOv3	Darknet-53	**72.2**	29.2	74.0	78.6	31.2	69.7	26.9	48.6	54.4	31.1	61.1	44.9	49.7	**87.4**	70.6	**68.7**	**87.3**	29.4	**48.3**	78.7	57.1
Faster RCNN with FPN	ResNet-50	54.1	71.4	63.3	81.0	42.6	72.5	57.5	68.7	62.1	73.1	76.5	42.8	56.0	71.8	57.0	53.5	81.2	53.0	43.1	80.9	63.1
	ResNet-101	54.0	74.5	63.3	80.7	44.8	72.5	60.0	75.6	62.3	76.0	76.8	46.4	57.2	71.8	68.3	53.8	81.1	59.5	43.1	81.2	65.1
Mask-RCNN with FPN	ResNet-50	53.8	72.3	63.2	81.0	38.7	72.6	55.9	71.6	67.0	73.0	75.8	44.2	56.5	71.9	58.6	53.6	81.1	54.0	43.1	81.1	63.5
	ResNet-101	53.9	76.6	63.2	80.9	40.2	72.5	60.4	76.3	62.5	76.0	75.9	46.5	57.4	71.8	68.3	53.7	81.0	**62.3**	43.0	81.0	65.2
RetinaNet	ResNet-50	53.7	77.3	69.0	81.3	44.1	72.3	62.5	76.2	66.0	77.7	74.2	50.7	59.6	71.2	69.3	44.8	81.3	54.2	45.1	83.4	65.7
	ResNet-101	53.3	77.0	69.3	**85.0**	44.1	73.2	62.4	78.6	62.8	78.6	76.6	49.9	59.6	71.1	68.4	45.8	81.3	55.2	44.4	85.5	**66.1**
PANet	ResNet-50	61.9	70.4	71.0	80.4	38.9	72.5	56.6	68.4	60.0	69.0	74.6	41.6	55.8	71.7	72.9	62.3	81.2	54.6	48.2	**86.7**	63.8
	ResNet-101	60.2	72.0	70.6	80.5	43.6	72.3	61.4	72.1	66.7	72.0	73.4	45.3	56.9	71.7	70.4	62.0	80.9	57.0	47.2	84.5	**66.1**
CornerNet	Hourglass-104	58.8	**84.2**	72.0	80.8	**46.4**	**75.3**	**64.3**	**81.6**	**76.3**	**79.5**	**79.5**	26.1	**60.6**	37.6	70.7	45.2	84.0	57.1	43.0	75.9	64.9

Keypoints from DIOR paper

1. **RetinaNet, PANet**
   - Highest overall mAP@0.50 of 66.1%
2. **YOLOv3**
   - Overall mAP@0.50 of 57.1%
   - Highest per class AP for
     - Airplane : 72.2%
     - Ship : 87.4%
     - Storage tank : 68.7%
     - Vehicle : 87.3 %
     - Tennis Court : 48.3%

c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
Airplane	Airport	Baseball field	Basketball court	Bridge	Chimney	Dam	Expressway service area	Expressway toll station	Golf course
c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
Ground track field	Harbor	Overpass	Ship	Stadium	Storage tank	Tennis court	Train station	Vehicle	Wind mill

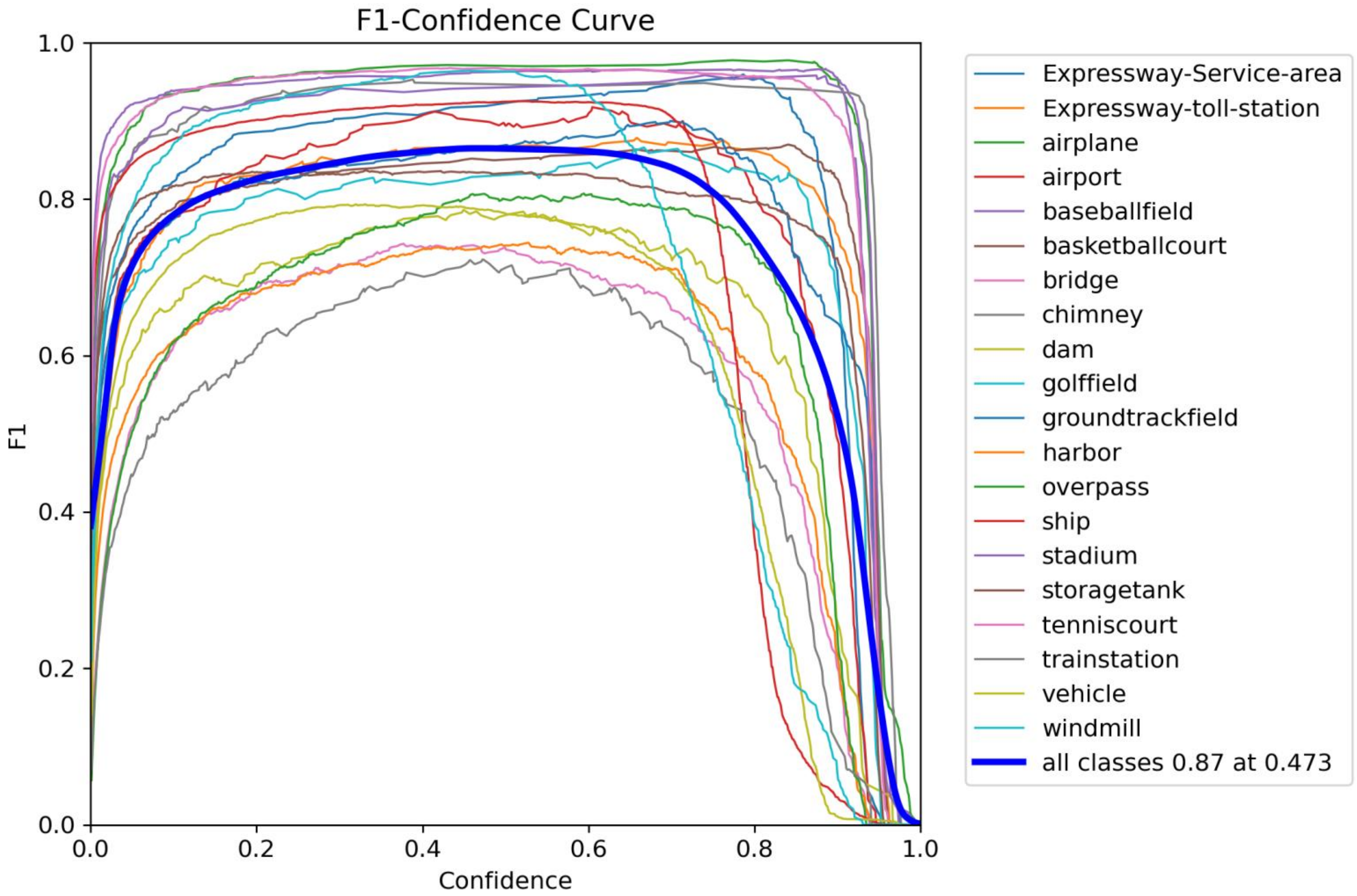
	Backbone	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	mAP
YOLOv3	Darknet-53	**72.2**	29.2	74.0	78.6	31.2	69.7	26.9	48.6	54.4	31.1	61.1	44.9	49.7	**87.4**	70.6	**68.7**	**87.3**	29.4	**48.3**	78.7	57.1
Exp 07 YOLOv8 Med Adam 0.01		86.9	90.9	87.4	91.7	54.5	80.4	76.3	92.3	76.5	85.4	85.5	70	67.4	92.9	82.3	81.4	93.5	68.6	57.6	90.7	80.6
Yolo-NAS Large SGD		62.31	49.48	60.51	62.79	66.55	81.37	25.35	64.75	28.26	64.31	65.87	45.99	41.84	55.83	56.94	50.64	79.32	29.31	31.38	41.88	53.23

References

- Ke Li *et. al.*, *"***Object Detection in Optical Remote Sensing Images: A Survey and A New Benchmark"** *ISPRS journal of photogrammetry and remote sensing* (2020). DOI: 10.1016/j.isprsjprs.2019.11.023
- Dilion Reis *et. al., "***Real-Time Flying Object Detection with YOLOv8**", DOI: https://doi.org/10.48550/arXiv.2305.09972
- https://github.com/Deci-AI/super-gradients
- https://roboflow.com/model/yolo-nas
- Ashish Vaswani *et. Al. "***Attention Is All You Need"** https://arxiv.org/abs/1706.03762
- https://medium.com/@shetesamrudhi509/yolo-nas-model-architecture-ede458f752bf

Supplementary Slides

Experimental 07: Training metrics



Experimental 07: Training metrics