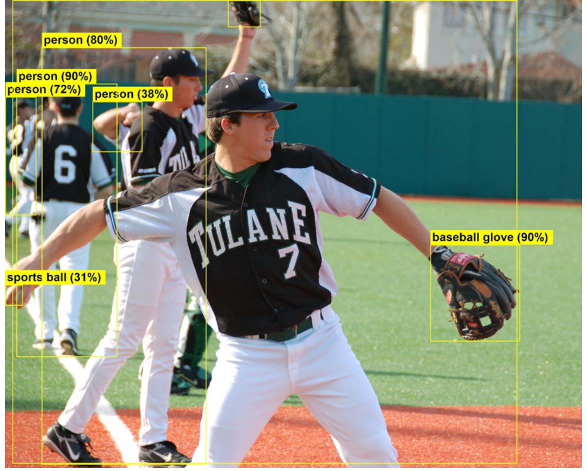
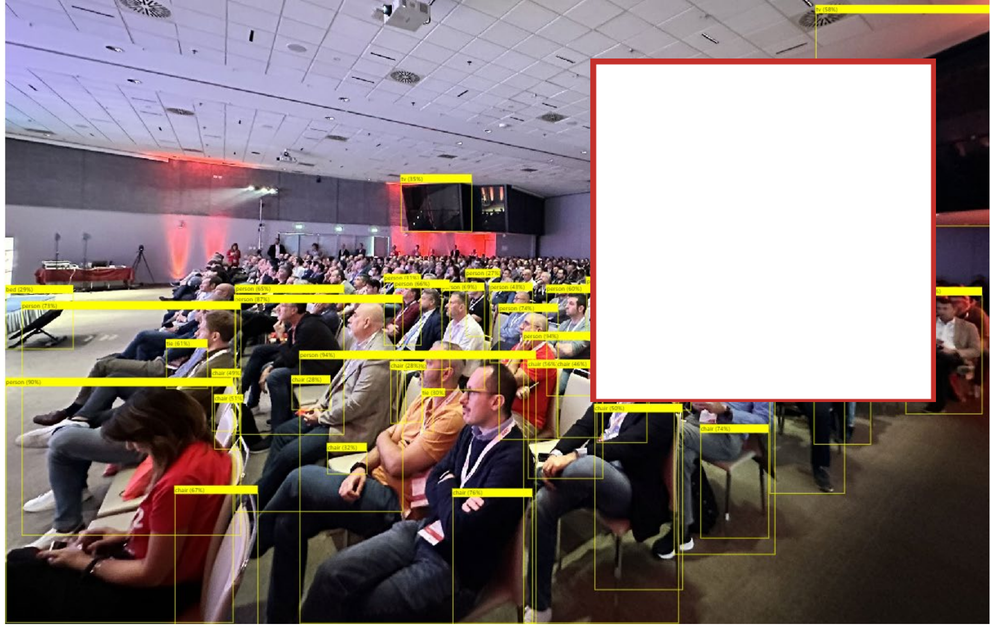
End-to-End Object Detection with Azure ML & ONNX Runtime

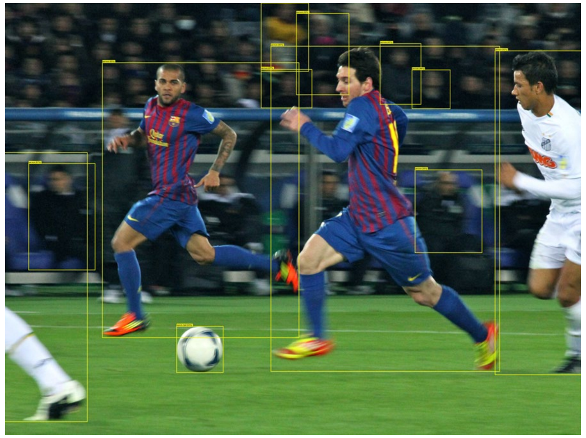
Clemente Giorio
Gianni Rosa Gallina

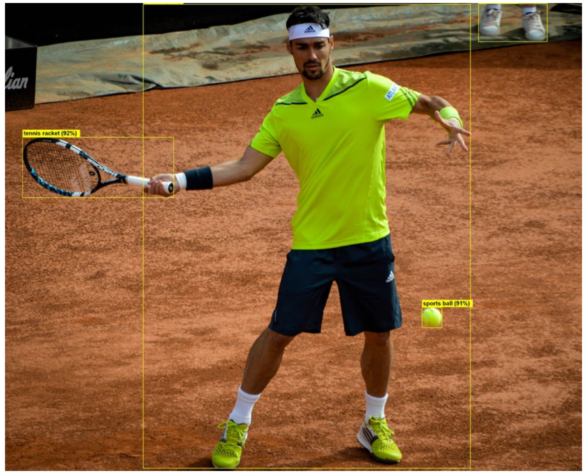
End-to-End Object Detection with Azure ML & ONNX Runtime

Object Detection







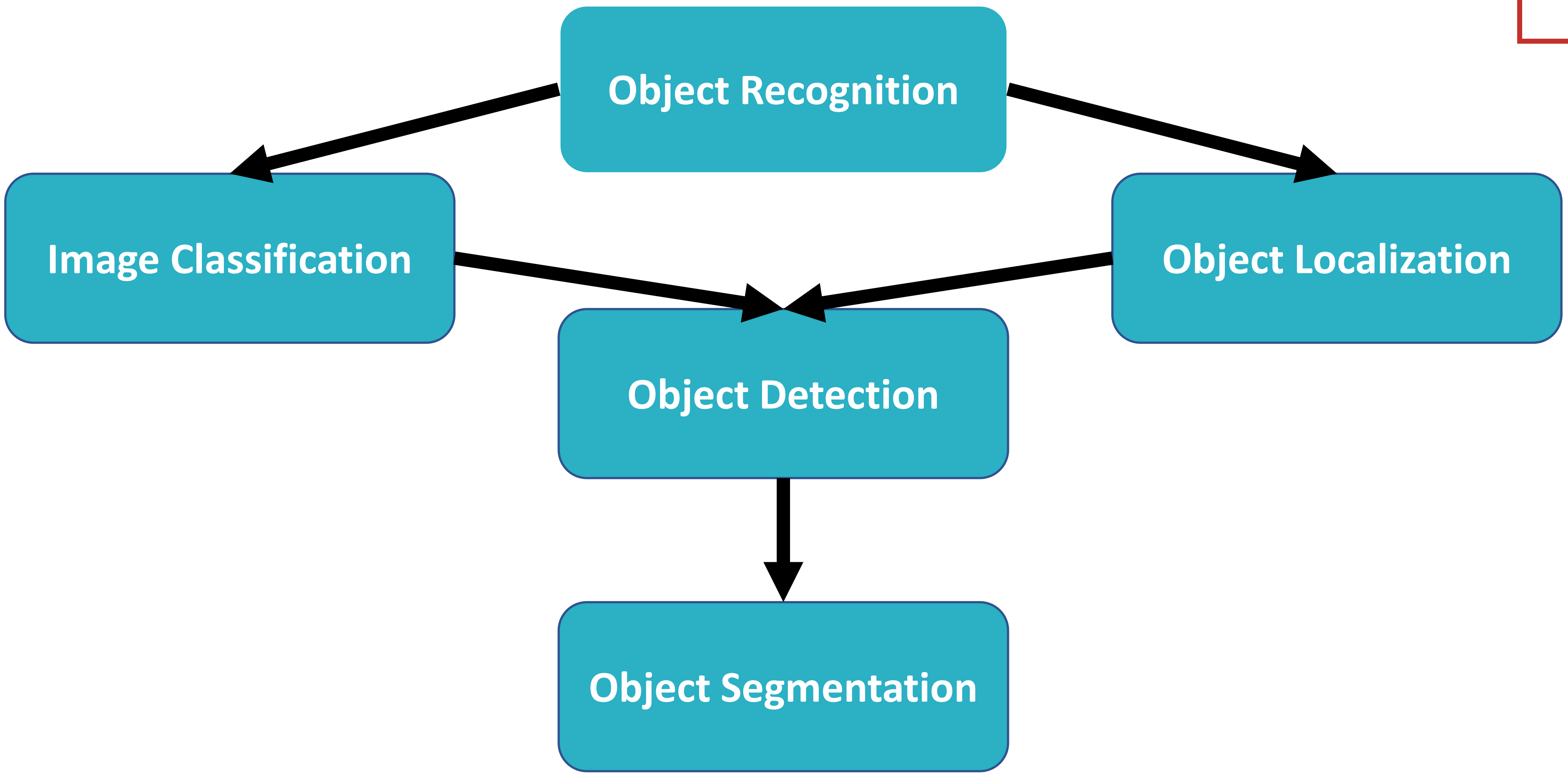


WPC22 27°EDIZIONE • MILANO

Source: Images from Wikimedia, twitter

 OVERNET.

Overview of Object Detection Vision Task



Overview of Object Detection Vision Task

Image Classification

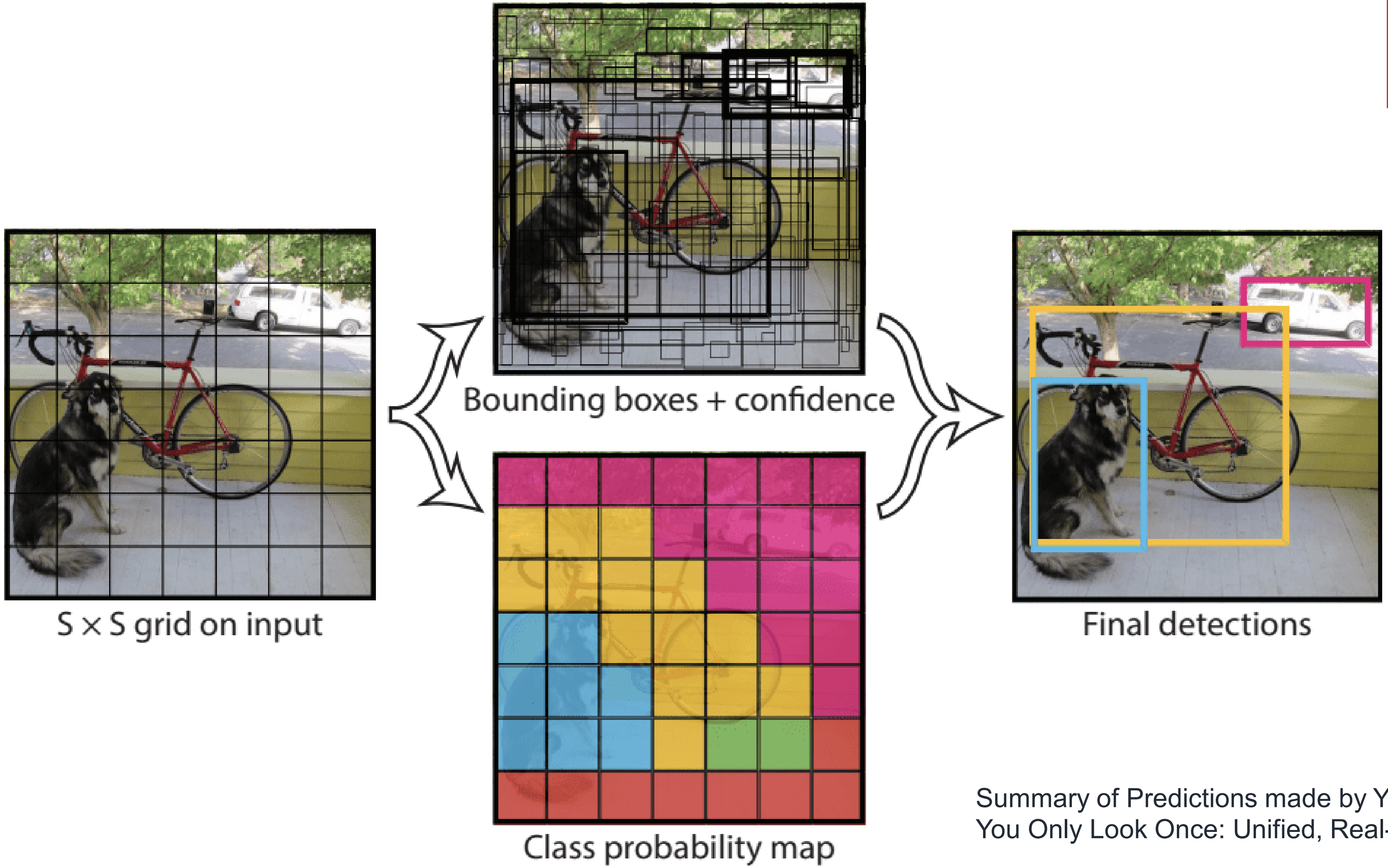
Output:
Class – Speaker

Object Detection

Output:
Bounding box – pc,x,y,w,h
Class – Speaker, Panza,Chair
Confidence – 0.94, 0.93, 0.85, and 0.88

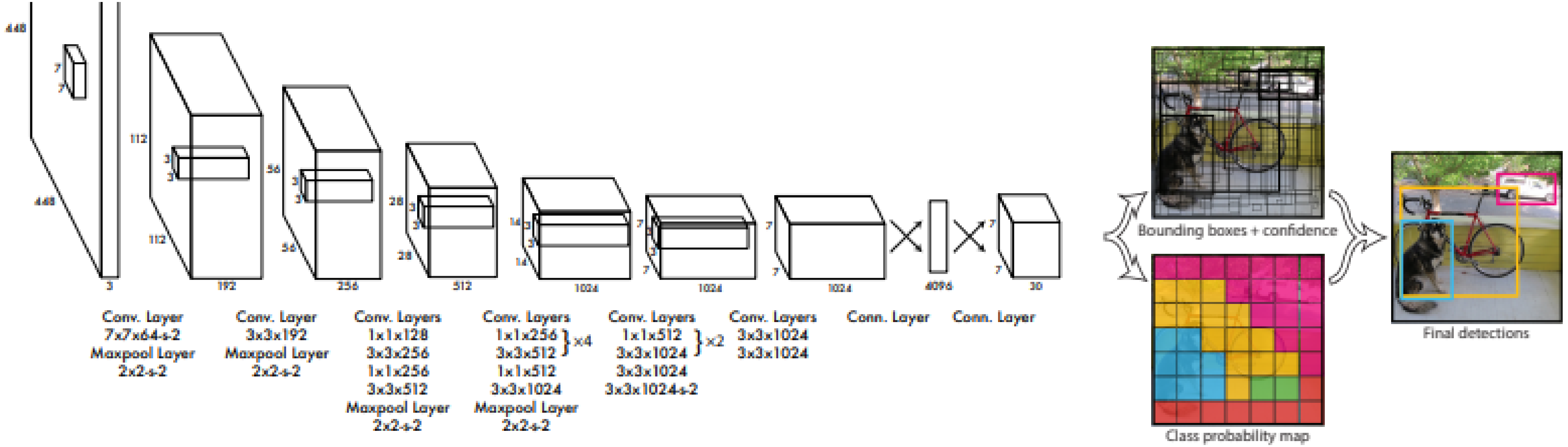
Source: Images from wikimedia

YOLO



Summary of Predictions made by YOLO Model.Taken from:
You Only Look Once: Unified, Real-Time Object Detection

YOLO



Summary of Predictions made by YOLO Model.Taken from:
You Look Only Once: Unified, Real-Time Object Detection

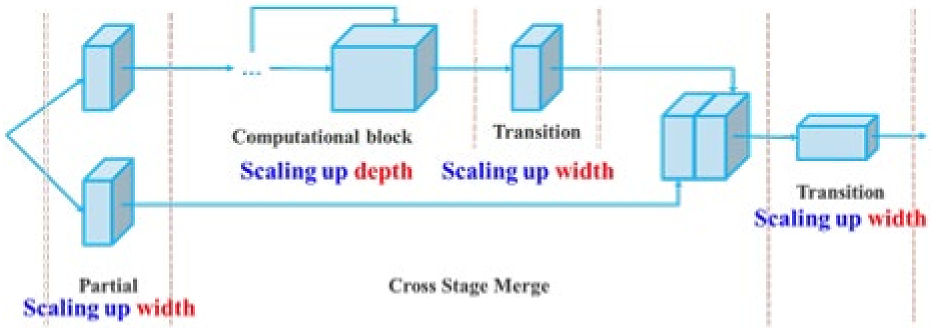
YOLO Architecture in general

Layers	YOLOv3	YOLOv4	YOLOv5
Neural Network Type	FCNN	FCNN	FCNN
Backbone	**Darknet53**	**CSPDarknet53** (CSPNet in Darknet)	**CSPDarkent53** Focus structure
Neck	**FPN** (Feature Pyramid Network)	**SPP** (Spatial Pyramid Pooling) and **PANet** (Path Aggregation Network)	**PANet**
Head	**B x (5 + C)** output layer **B**: No. of bounding boxes **C**: Class score	Same as Yolo v3	Same as Yolo v3
Loss Function	Binary Cross Entropy	Binary Cross Entropy	Binary Cross Entropy and Logit Loss Function

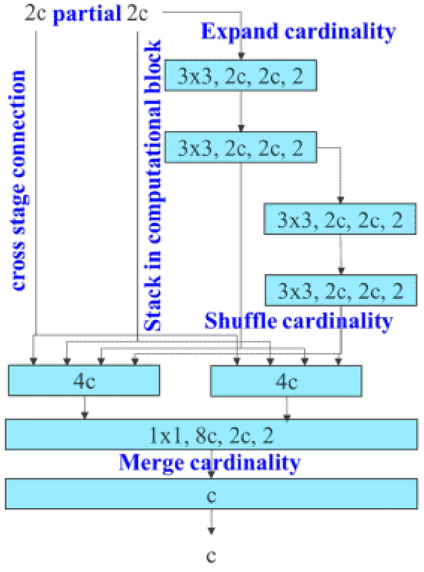
Source: https://learnopencv.com/wp-content/uploads/2022/04/model-architecture-yolo-summary.jpg

YOLOv7 Architecture

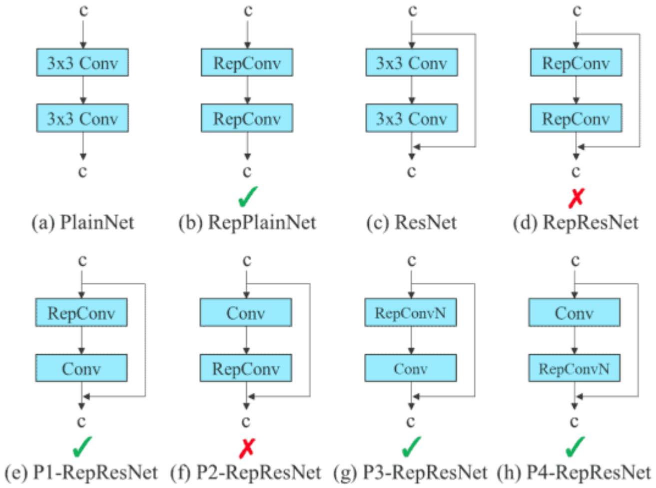
- **Architectural Reforms**
  - E-ELAN (Extended Efficient Layer Aggregation Network)
  - Model Scaling for Concatenation-based Models
- **Trainable BoF (Bag of Freebies)**
  - Planned re-parameterized convolution
  - Coarse for auxiliary and Fine for lead loss

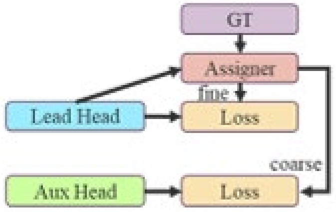


(c) compound scaling up depth and width for concatenation-based model



(d) E-ELAN

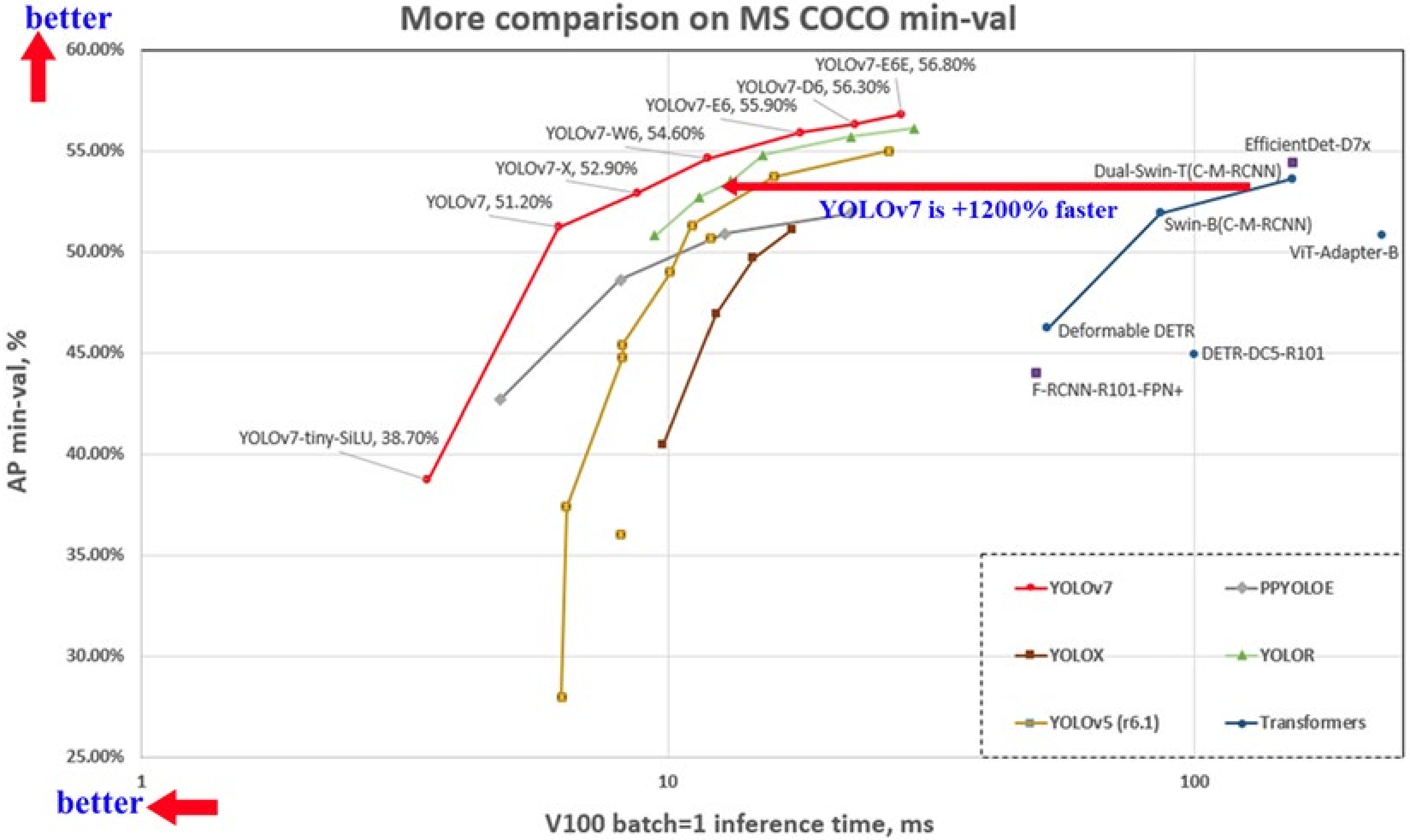




Coarse-to-fine lead guided assigner

Source: https://arxiv.org/pdf/2207.02696.pdf

YOLOv7 Performance



Source: https://arxiv.org/pdf/2207.02696.pdf

Comparison of baseline object detectors

Model	#Param.	FLOPs	Size	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	**51.2%**	**69.7%**	**55.5%**	**35.2%**	**56.0%**	**66.7%**
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X [81]	96.9M	226.8G	640	52.7%	**71.3%**	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	**52.9%**	71.1%	**57.5%**	**36.9%**	**57.7%**	**68.6%**
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	**35.2%**	**52.8%**	**37.3%**	**15.7%**	**38.0%**	**53.4%**
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	**10.9%**	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	**30.8%**	**47.3%**	**32.2%**	10.0%	**31.9%**	**52.2%**
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	**60.4%**	69.2%
YOLOv7-E6	97.2M	515.2G	1280	**55.9%**	**73.5%**	**61.1%**	**40.6%**	60.3%	**70.0%**
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	**42.4%**	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	**56.8%**	**74.4%**	**62.1%**	40.8%	**62.1%**	**70.6%**
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

Source: https://arxiv.org/pdf/2207.02696.pdf

Comprarison of state-of-art real-time object detector

Model	#Param.	FLOPs	Size	FPS	AP^{test} / AP^{val}	AP_{50}^{test}	AP_{75}^{test}	AP_{S}^{test}	AP_{M}^{test}	AP_{L}^{test}
YOLOX-S [21]	9.0M	26.8G	640	102	40.5% / 40.5%	-	-	-	-	-
YOLOX-M [21]	25.3M	73.8G	640	81	47.2% / 46.9%	-	-	-	-	-
YOLOX-L [21]	54.2M	155.6G	640	69	50.1% / 49.7%	-	-	-	-	-
YOLOX-X [21]	99.1M	281.9G	640	58	51.5% / 51.1%	-	-	-	-	-
PPYOLOE-S [85]	7.9M	17.4G	640	208	43.1% / 42.7%	60.5%	46.6%	23.2%	46.4%	56.9%
PPYOLOE-M [85]	23.4M	49.9G	640	123	48.9% / 48.6%	66.5%	53.0%	28.6%	52.9%	63.8%
PPYOLOE-L [85]	52.2M	110.1G	640	78	51.4% / 50.9%	68.9%	55.6%	31.4%	55.3%	66.1%
PPYOLOE-X [85]	98.4M	206.6G	640	45	52.2% / 51.9%	69.9%	56.5%	33.3%	56.3%	66.4%
YOLOv5-N (r6.1) [23]	1.9M	4.5G	640	159	- / 28.0%	-	-	-	-	-
YOLOv5-S (r6.1) [23]	7.2M	16.5G	640	156	- / 37.4%	-	-	-	-	-
YOLOv5-M (r6.1) [23]	21.2M	49.0G	640	122	- / 45.4%	-	-	-	-	-
YOLOv5-L (r6.1) [23]	46.5M	109.1G	640	99	- / 49.0%	-	-	-	-	-
YOLOv5-X (r6.1) [23]	86.7M	205.7G	640	83	- / 50.7%	-	-	-	-	-
YOLOR-CSP [81]	52.9M	120.4G	640	106	51.1% / 50.8%	69.6%	55.7%	31.7%	55.3%	64.7%
YOLOR-CSP-X [81]	96.9M	226.8G	640	87	53.0% / 52.7%	71.4%	57.9%	33.7%	57.1%	66.8%
YOLOv7-tiny-SiLU	6.2M	13.8G	640	286	38.7% / 38.7%	56.7%	41.7%	18.8%	42.4%	51.9%
YOLOv7	36.9M	104.7G	640	161	51.4% / 51.2%	69.7%	55.9%	31.8%	55.5%	65.0%
YOLOv7-X	71.3M	189.9G	640	114	53.1% / 52.9%	71.2%	57.8%	33.8%	57.1%	67.4%
YOLOv5-N6 (r6.1) [23]	3.2M	18.4G	1280	123	- / 36.0%	-	-	-	-	-
YOLOv5-S6 (r6.1) [23]	12.6M	67.2G	1280	122	- / 44.8%	-	-	-	-	-
YOLOv5-M6 (r6.1) [23]	35.7M	200.0G	1280	90	- / 51.3%	-	-	-	-	-
YOLOv5-L6 (r6.1) [23]	76.8M	445.6G	1280	63	- / 53.7%	-	-	-	-	-
YOLOv5-X6 (r6.1) [23]	140.7M	839.2G	1280	38	- / 55.0%	-	-	-	-	-
YOLOR-P6 [81]	37.2M	325.6G	1280	76	53.9% / 53.5%	71.4%	58.9%	36.1%	57.7%	65.6%
YOLOR-W6 [81]	79.8G	453.2G	1280	66	55.2% / 54.8%	72.7%	60.5%	37.7%	59.1%	67.1%
YOLOR-E6 [81]	115.8M	683.2G	1280	45	55.8% / 55.7%	73.4%	61.1%	38.4%	59.7%	67.7%
YOLOR-D6 [81]	151.7M	935.6G	1280	34	56.5% / 56.1%	74.1%	61.9%	38.9%	60.4%	68.7%
YOLOv7-W6	70.4M	360.0G	1280	84	54.9% / 54.6%	72.6%	60.1%	37.3%	58.7%	67.1%
YOLOv7-E6	97.2M	515.2G	1280	56	56.0% / 55.9%	73.5%	61.2%	38.0%	59.9%	68.4%
YOLOv7-D6	154.7M	806.8G	1280	44	56.6% / 56.3%	74.0%	61.8%	38.8%	60.1%	69.5%
YOLOv7-E6E	151.7M	843.2G	1280	36	56.8% / 56.8%	74.4%	62.1%	39.3%	60.5%	69.0%

Source: https://arxiv.org/pdf/2207.02696.pdf

Model	#Param.	FLOPs	Size	FPS^{V100}	AP^{test} / AP^{val}	AP$_{50}^{test}$	AP$_{75}^{test}$
YOLOv7-tiny-SiLU	6.2M	13.8G	640	286	**38.7% / 38.7%**	**56.7%**	**41.7%**
PPYOLOE-S [85]	7.9M	17.4G	640	208	**43.1% / 42.7%**	**60.5%**	**46.6%**
YOLOv7	36.9M	104.7G	640	161	**51.4% / 51.2%**	**69.7%**	**55.9%**
YOLOv5-N (r6.1) [23]	1.9M	4.5G	640	159	- / 28.0%	-	-
YOLOv5-S (r6.1) [23]	7.2M	16.5G	640	156	- / 37.4%	-	-
PPYOLOE-M [85]	23.4M	49.9G	640	123	48.9% / 48.6%	66.5%	53.0%
YOLOv5-N6 (r6.1) [23]	3.2M	18.4G	1280	123	- / 36.0%	-	-
YOLOv5-S6 (r6.1) [23]	12.6M	67.2G	1280	122	- / 44.8%	-	-
YOLOv5-M (r6.1) [23]	21.2M	49.0G	640	122	- / 45.4%	-	-
YOLOv7-X	71.3M	189.9G	640	114	**53.1% / 52.9%**	**71.2%**	**57.8%**
YOLOR-CSP [81]	52.9M	120.4G	640	106	51.1% / 50.8%	69.6%	55.7%
YOLOX-S [21]	9.0M	26.8G	640	102	40.5% / 40.5%	-	-
YOLOv5-L (r6.1) [23]	46.5M	109.1G	640	99	- / 49.0%	-	-
YOLOv5-M6 (r6.1) [23]	35.7M	200.0G	1280	90	- / 51.3%	-	-
YOLOR-CSP-X [81]	96.9M	226.8G	640	87	53.0% / 52.7%	**71.4%**	**57.9%**
YOLOv7-W6	70.4M	360.0G	1280	84	**54.9% / 54.6%**	**72.6%**	**60.1%**
YOLOv5-X (r6.1) [23]	86.7M	205.7G	640	83	- / 50.7%	-	-
YOLOX-M [21]	25.3M	73.8G	640	81	47.2% / 46.9%	-	-
PPYOLOE-L [85]	52.2M	110.1G	640	78	51.4% / 50.9%	68.9%	55.6%
YOLOR-P6 [81]	37.2M	325.6G	1280	76	53.9% / 53.5%	71.4%	58.9%
YOLOX-L [21]	54.2M	155.6G	640	69	50.1% / 49.7%	-	-
YOLOR-W6 [81]	79.8G	453.2G	1280	66	**55.2% / 54.8%**	**72.7%**	**60.5%**
YOLOv5-L6 (r6.1) [23]	76.8M	445.6G	1280	63	- / 53.7%	-	-
YOLOX-X [21]	99.1M	281.9G	640	58	51.5% / 51.1%	-	-
YOLOv7-E6	97.2M	515.2G	1280	56	**56.0% / 55.9%**	**73.5%**	**61.2%**
YOLOR-E6 [81]	115.8M	683.2G	1280	45	55.8% / 55.7%	73.4%	61.1%
PPYOLOE-X [85]	98.4M	206.6G	640	45	52.2% / 51.9%	69.9%	56.5%
YOLOv7-D6	154.7M	806.8G	1280	44	**56.6% / 56.3%**	**74.0%**	**61.8%**
YOLOv5-X6 (r6.1) [23]	140.7M	839.2G	1280	38	- / 55.0%	-	-
YOLOv7-E6E	151.7M	843.2G	1280	36	**56.8% / 56.8%**	**74.4%**	**62.1%**
YOLOR-D6 [81]	151.7M	935.6G	1280	34	56.5% / 56.1%	**74.1%**	**61.9%**

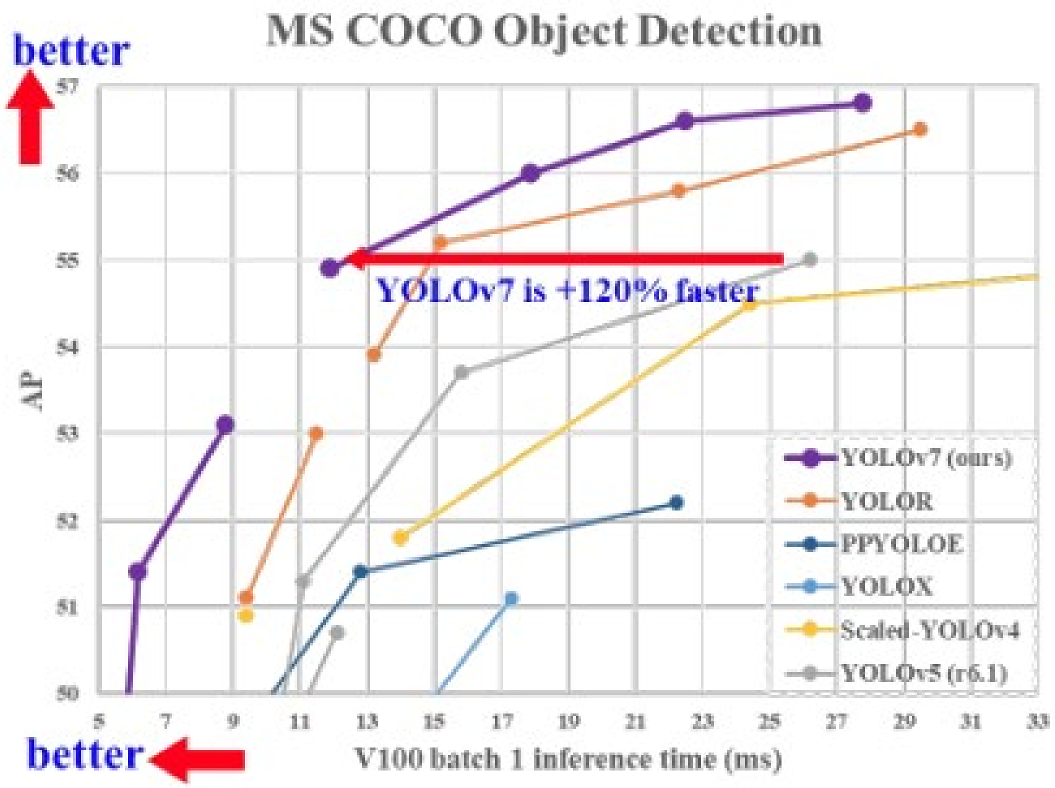
Source: https://arxiv.org/pdf/2207.02696.pdf

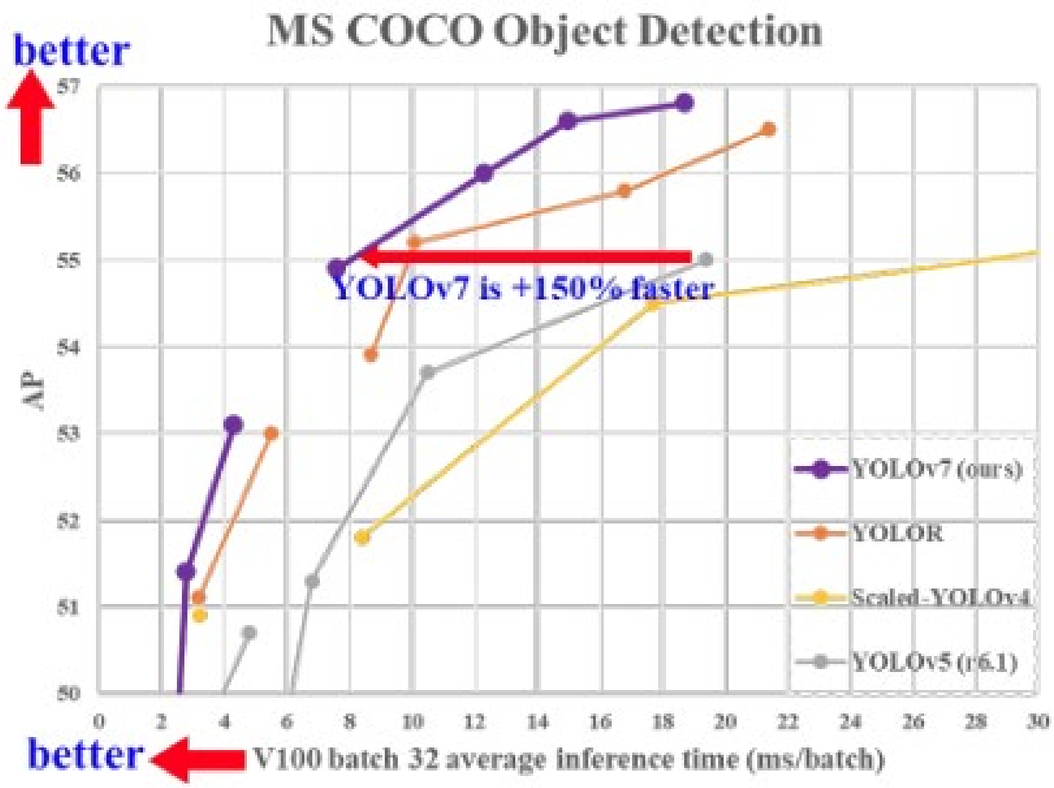
YOLOv5-L (r6.1) [23]	46.5M	109.1G	640	99	- / 49.0%	-	-
YOLOv5-M6 (r6.1) [23]	35.7M	200.0G	1280	90	- / 51.3%	-	-
YOLOR-CSP-X [81]	96.9M	226.8G	640	87	53.0% / 52.7%	**71.4%**	**57.9%**
YOLOv7-W6	70.4M	360.0G	1280	84	**54.9% / 54.6%**	**72.6%**	**60.1%**
YOLOv5-X (r6.1) [23]	86.7M	205.7G	640	83	- / 50.7%	-	-
YOLOX-M [21]	25.3M	73.8G	640	81	47.2% / 46.9%	-	-
PPYOLOE-L [85]	52.2M	110.1G	640	78	51.4% / 50.9%	68.9%	55.6%
YOLOR-P6 [81]	37.2M	325.6G	1280	76	53.9% / 53.5%	71.4%	58.9%
YOLOX-L [21]	54.2M	155.6G	640	69	50.1% / 49.7%	-	-
YOLOR-W6 [81]	79.8G	453.2G	1280	66	**55.2% / 54.8%**	**72.7%**	**60.5%**
YOLOv5-L6 (r6.1) [23]	76.8M	445.6G	1280	63	- / 53.7%	-	-
YOLOX-X [21]	99.1M	281.9G	640	58	51.5% / 51.1%	-	-
YOLOv7-E6	97.2M	515.2G	1280	56	**56.0% / 55.9%**	**73.5%**	**61.2%**
YOLOR-E6 [81]	115.8M	683.2G	1280	45	55.8% / 55.7%	73.4%	61.1%
PPYOLOE-X [85]	98.4M	206.6G	640	45	52.2% / 51.9%	69.9%	56.5%
YOLOv7-D6	154.7M	806.8G	1280	44	**56.6% / 56.3%**	**74.0%**	**61.8%**
YOLOv5-X6 (r6.1) [23]	140.7M	839.2G	1280	38	- / 55.0%	-	-
YOLOv7-E6E	151.7M	843.2G	1280	36	**56.8% / 56.8%**	**74.4%**	**62.1%**
YOLOR-D6 [81]	151.7M	935.6G	1280	34	56.5% / 56.1%	74.1%	61.9%
F-RCNN-R101-FPN+ [5]	60.0M	246.0G	1333	20	- / 44.0%	-	-
Deformable DETR [100]	40.0M	173.0G	-	19	- / 46.2%	-	-
Swin-B (C-M-RCNN) [52]	145.0M	982.0G	1333	11.6	- / 51.9%	-	-
DETR DC5-R101 [5]	60.0M	253.0G	1333	10	- / 44.9%	-	-
EfficientDet-D7x [74]	77.0M	410.0G	1536	6.5	55.1% / 54.4%	72.4%	58.4%
Dual-Swin-T (C-M-RCNN) [47]	113.8M	836.0G	1333	6.5	- / 53.6%	-	-
ViT-Adapter-B [7]	122.0M	997.0G	-	4.4	- / 50.8%	-	-
Dual-Swin-B (HTC) [47]	235.0M	-	1600	2.5	**58.7% / 58.4%**	-	-
Dual-Swin-L (HTC) [47]	453.0M	-	1600	1.5	**59.4% / 59.1%**	-	-

Model	#Param.	FLOPs	Size	FPS^{A100}	AP^{test} / AP^{val}	AP_{50}^{test}	AP_{75}^{test}
DN-Deformable-DETR [41]	48.0M	265.0G	1333	23.0	- / 48.6%	-	-
ConvNeXt-B (C-M-RCNN) [53]	-	964.0G	1280	11.5	- / 54.0%	73.1%	58.8%
Swin-B (C-M-RCNN) [52]	-	982.0G	1280	10.7	- / 53.0%	71.8%	57.5%
DINO-5scale (R50) [89]	47.0M	860.0G	1333	10.0	- / 51.0%	-	-
ConvNeXt-L (C-M-RCNN) [53]	-	1354.0G	1280	10.0	- / 54.8%	73.8%	59.8%
Swin-L (C-M-RCNN) [52]	-	1382.0G	1280	9.2	- / 53.9%	72.4%	58.8%
ConvNeXt-XL (C-M-RCNN) [53]	-	1898.0G	1280	8.6	- / 55.2%	74.2%	59.9%

Source: https://arxiv.org/pdf/2207.02696.pdf

YOLOv7 performance





Source: https://arxiv.org/pdf/2207.02696.pdf

YOLOv7 – Pretrained Checkpoints

Model	Test Size	AP^{test}	AP_{50}^{test}	AP_{75}^{test}	batch 1 fps	batch 32 average time
YOLOv7	640	51.4%	69.7%	55.9%	161 *fps*	2.8 *ms*
YOLOv7-X	640	53.1%	71.2%	57.8%	114 *fps*	4.3 *ms*
YOLOv7-W6	1280	54.9%	72.6%	60.1%	84 *fps*	7.6 *ms*
YOLOv7-E6	1280	56.0%	73.5%	61.2%	56 *fps*	12.3 *ms*
YOLOv7-D6	1280	56.6%	74.0%	61.8%	44 *fps*	15.0 *ms*
YOLOv7-E6E	1280	56.8%	74.4%	62.1%	36 *fps*	18.7 *ms*

Source: https://github.com/WongKinYiu/yolov7

What is ML.NET and how does it work?

ML.NET

ML.NET

Sentiment analysis
Analyze the sentiment of customer reviews using a binary classification algorithm.

Product recommendation
Recommend products based on purchase history using a matrix factorization algorithm.

Price prediction
Predict taxi fares based on parameters such as distance traveled using a regression algorithm.

Customer segmentation
Identify groups of customers with similar profiles using a clustering algorithm.

Object detection
Recognize objects in an image using an ONNX deep learning model.

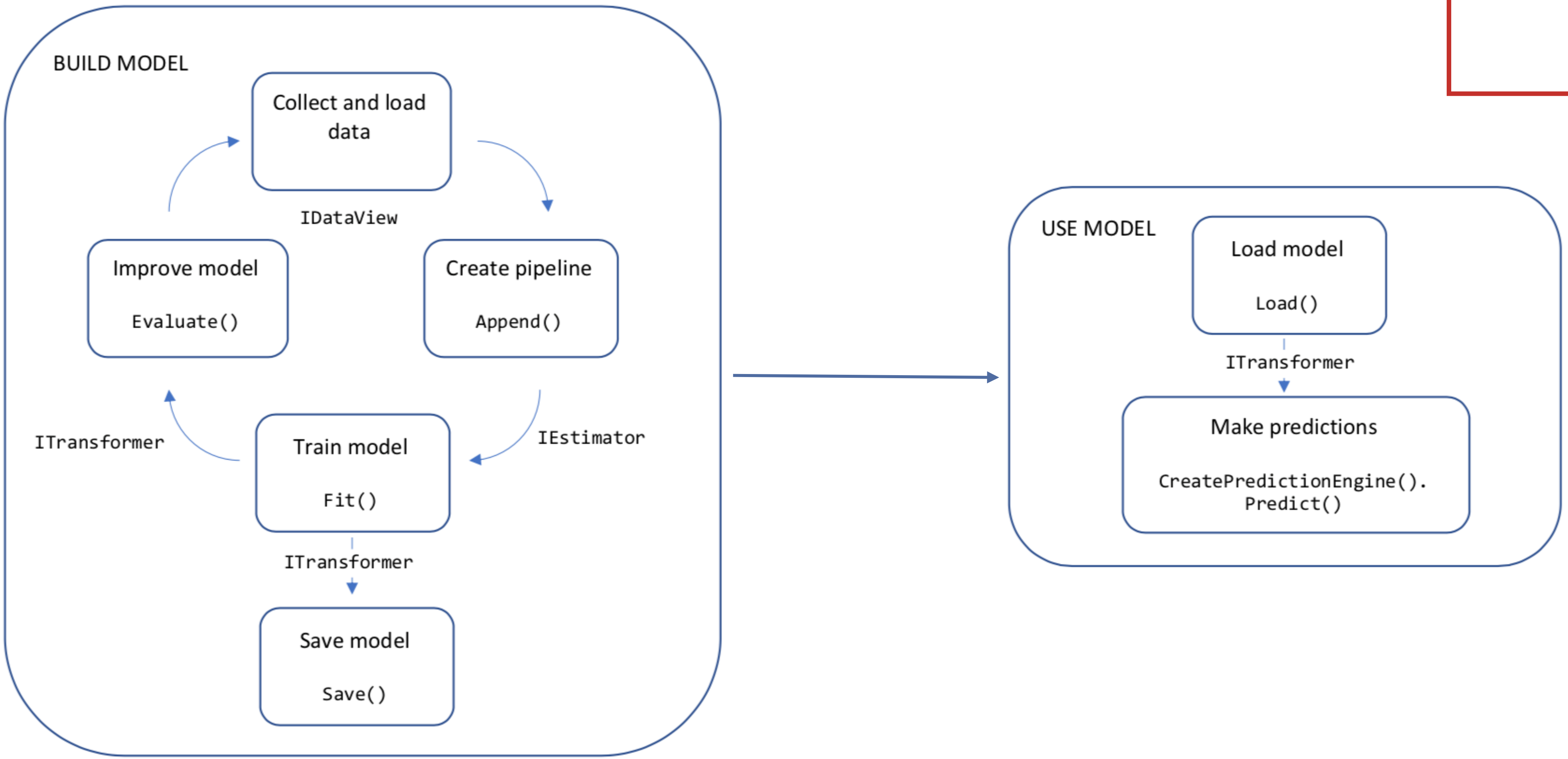
Fraud detection
Detect fraudulent credit card transactions using a binary classification algorithm.

Sales spike detection
Detect spikes and changes in product sales using an anomaly detection model.

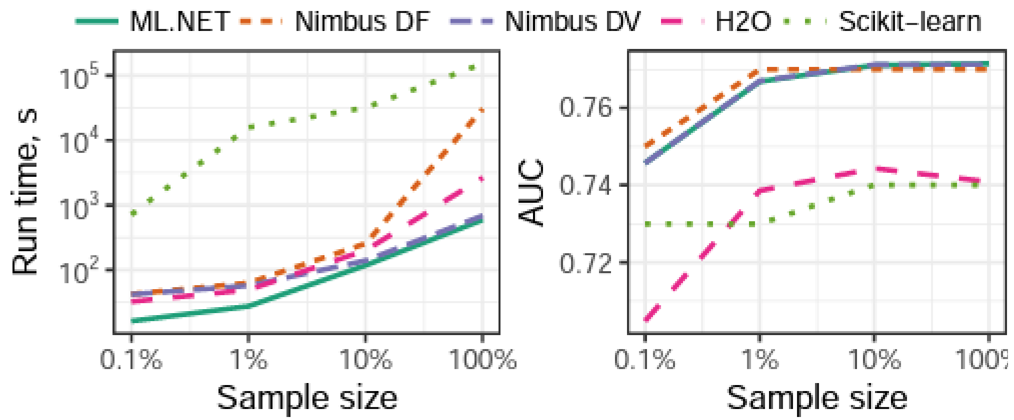
Image classification
Classify images (for example, broccoli vs. pizza) using a TensorFlow deep learning model.

Sales forecasting
Forecast future sales for products using a regression algorithm.

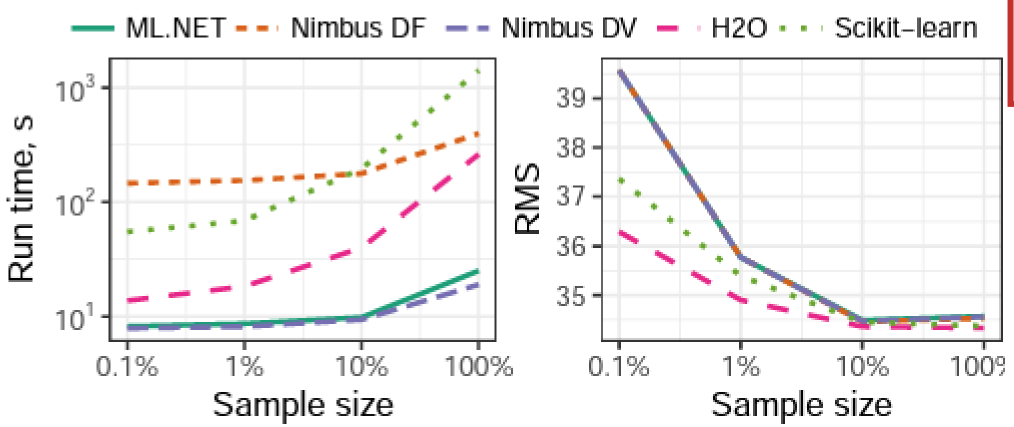
Code workflow



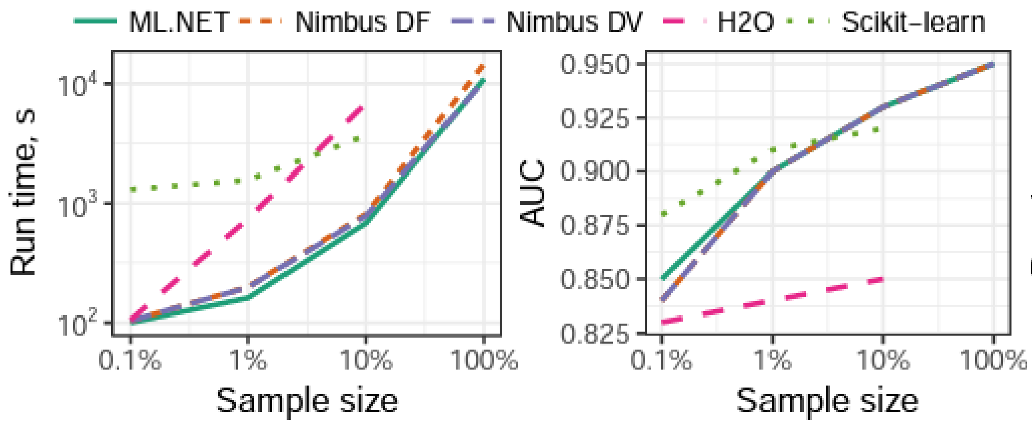
ML.NET



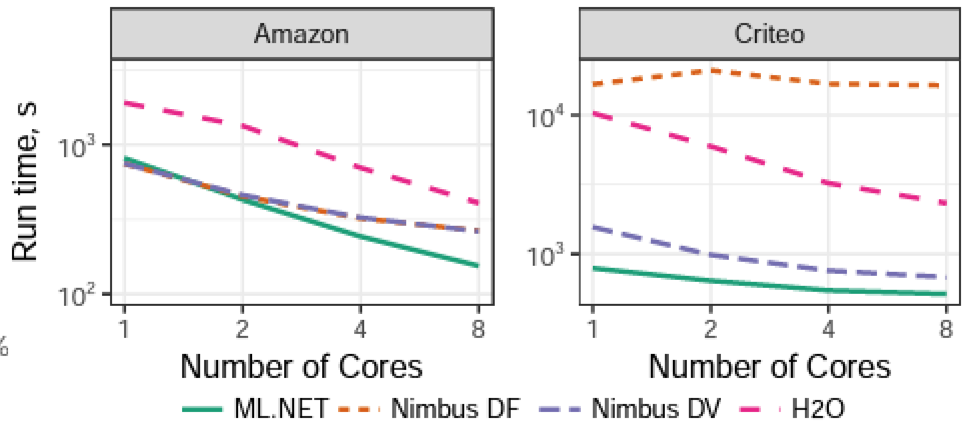
Experimental results for the Criteo dataset



Experimental results for the Flight Delay dataset



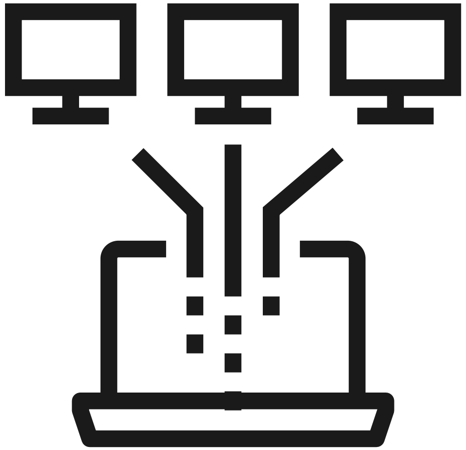
Experimental results for the Amazon dataset



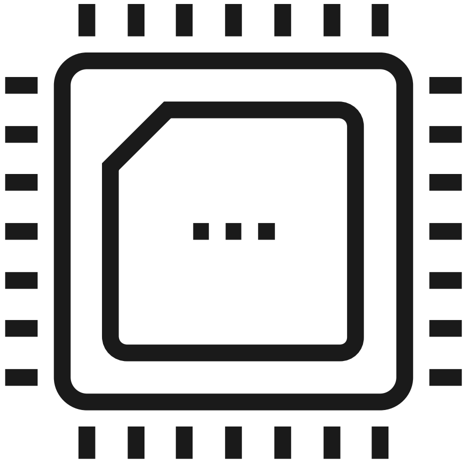
Scale-up experiments

ONNX

ONNX – Key Benefits

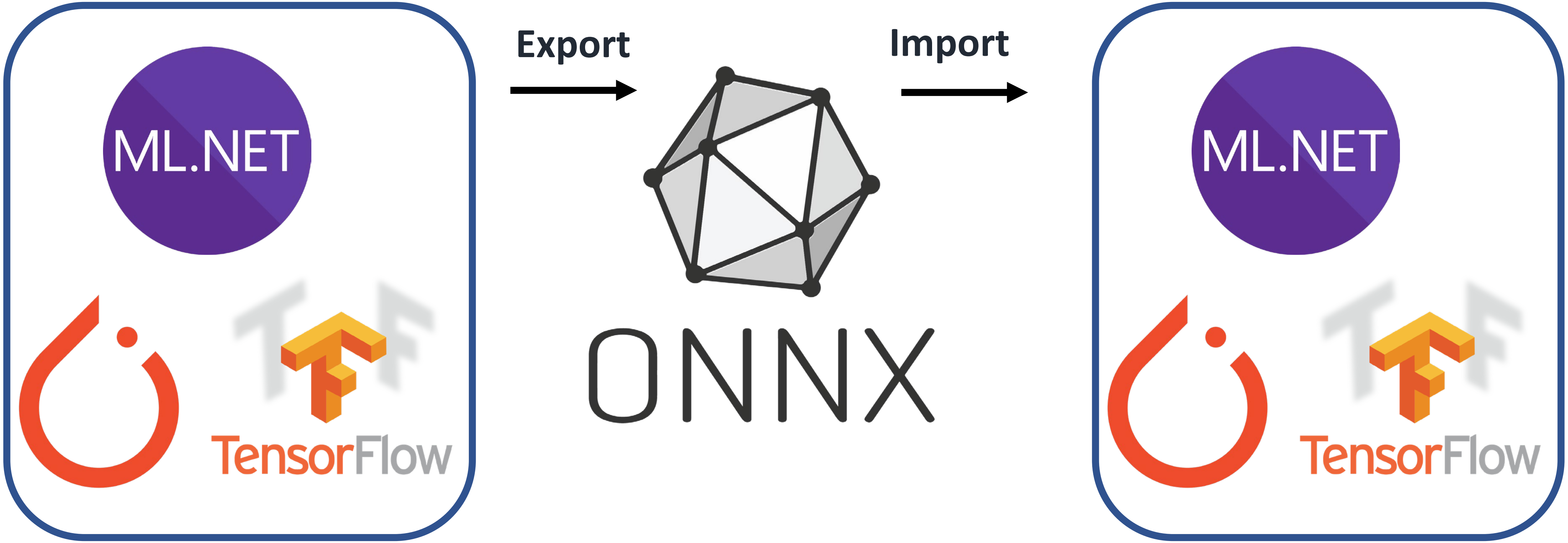


Interoperability



Hardware Access

ONNX



NETRON

https://github.com/lutzroeder/netron

Avalonia UI

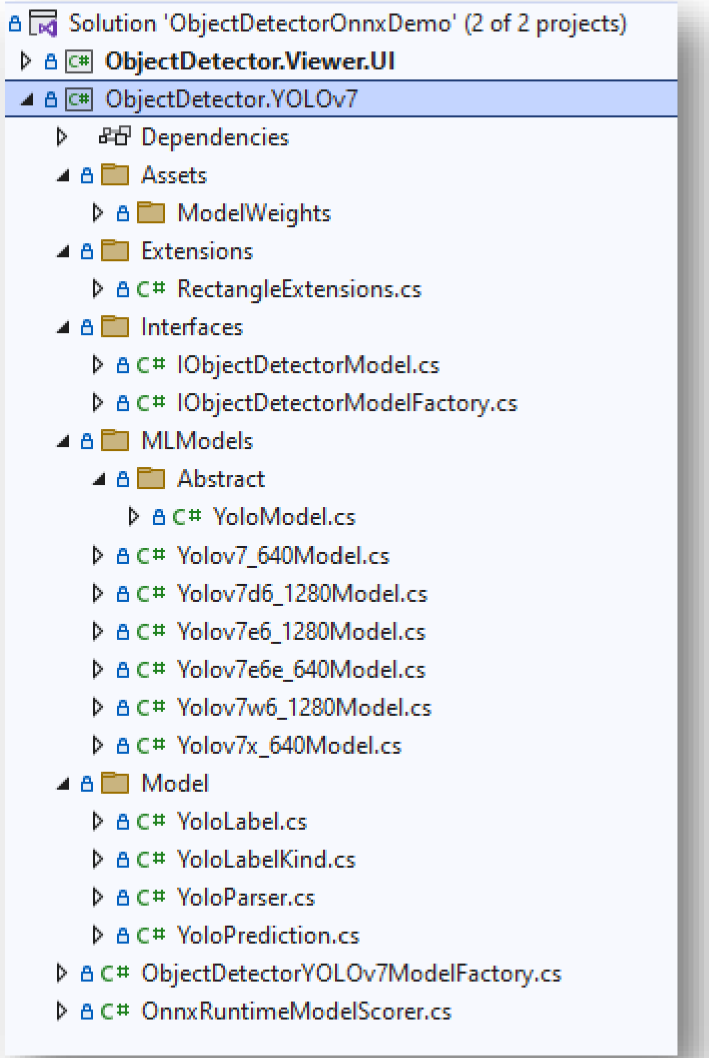
Open Source

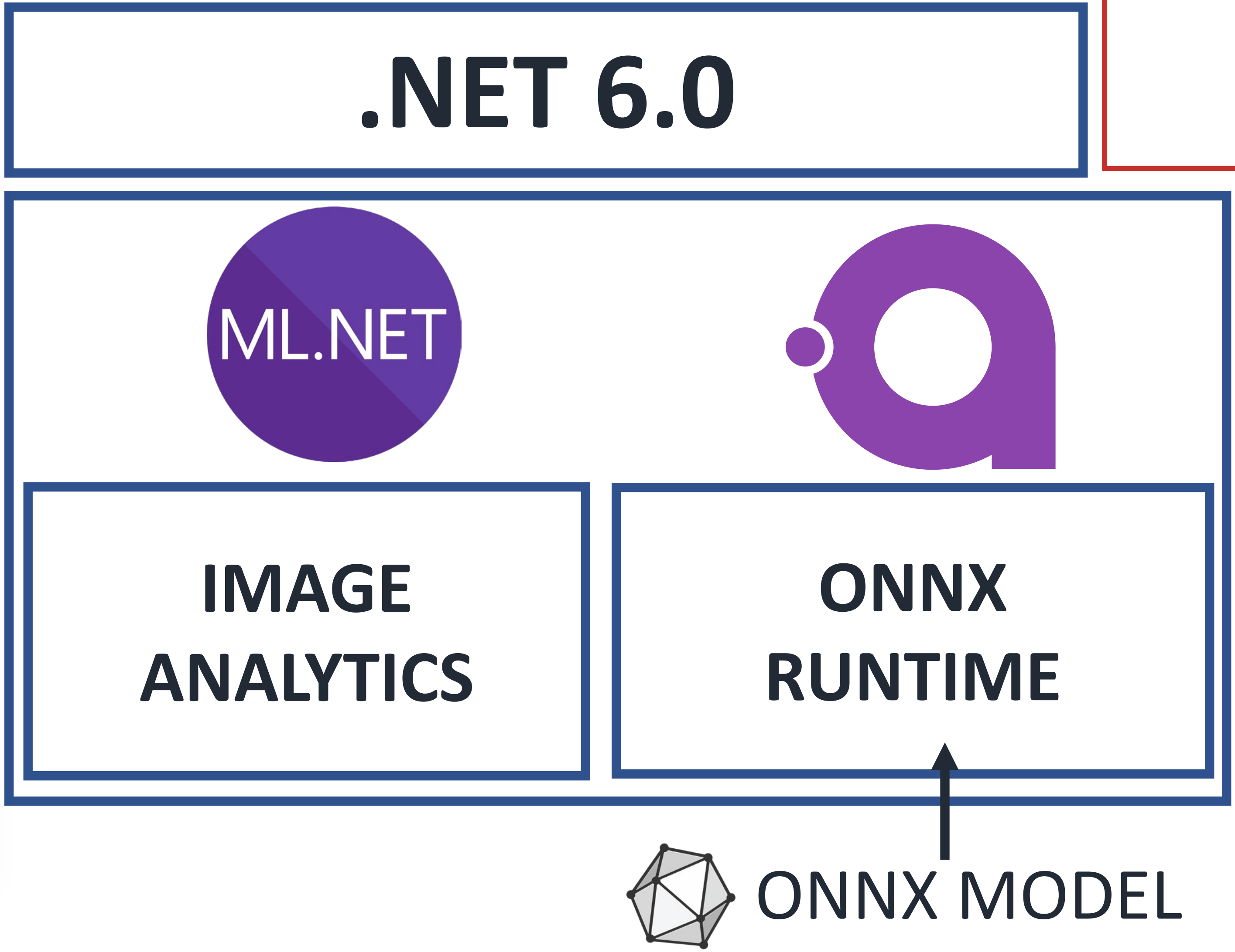
Visual Studio Extension

Open JetBrains Rider & #

Reuse Existing Skills

https://avaloniaui.net/


Solution 'ObjectDetectorOnnxDemo' (2 of 2 projects)
ObjectDetector.Viewer.UI
ObjectDetector.YOLOv7
Dependencies
Assets
ModelWeights
Extensions
RectangleExtensions.cs
Interfaces
IObjectDetectorModel.cs
IObjectDetectorModelFactory.cs
MLModels
Abstract
YoloModel.cs
Yolov7_640Model.cs
Yolov7d6_1280Model.cs
Yolov7e6_1280Model.cs
Yolov7e6e_640Model.cs
Yolov7w6_1280Model.cs
Yolov7x_640Model.cs
Model
YoloLabel.cs
YoloLabelKind.cs
YoloParser.cs
YoloPrediction.cs
ObjectDetectorYOLOv7ModelFactory.cs
OnnxRuntimeModelScorer.cs


.NET 6.0
ML.NET
IMAGE ANALYTICS
ONNX RUNTIME
ONNX MODEL

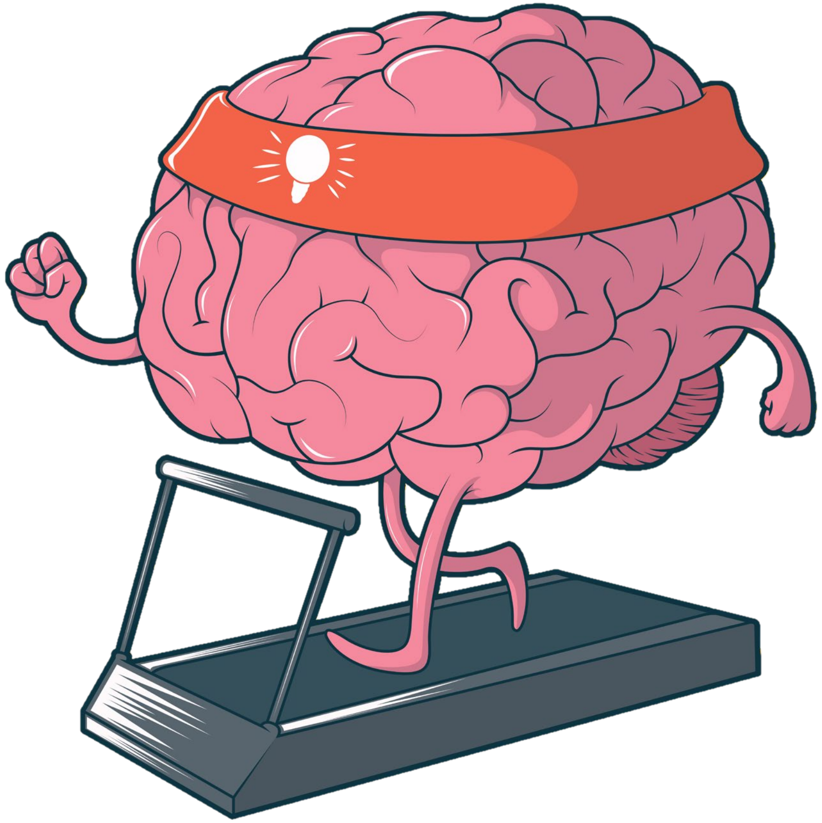
Other frameworks

TensorFlow PyTorch

TorchSharp: https://github.com/dotnet/TorchSharp

TensorFlow.NET: https://github.com/SciSharp/TensorFlow.NET

Training/Fine Tuning

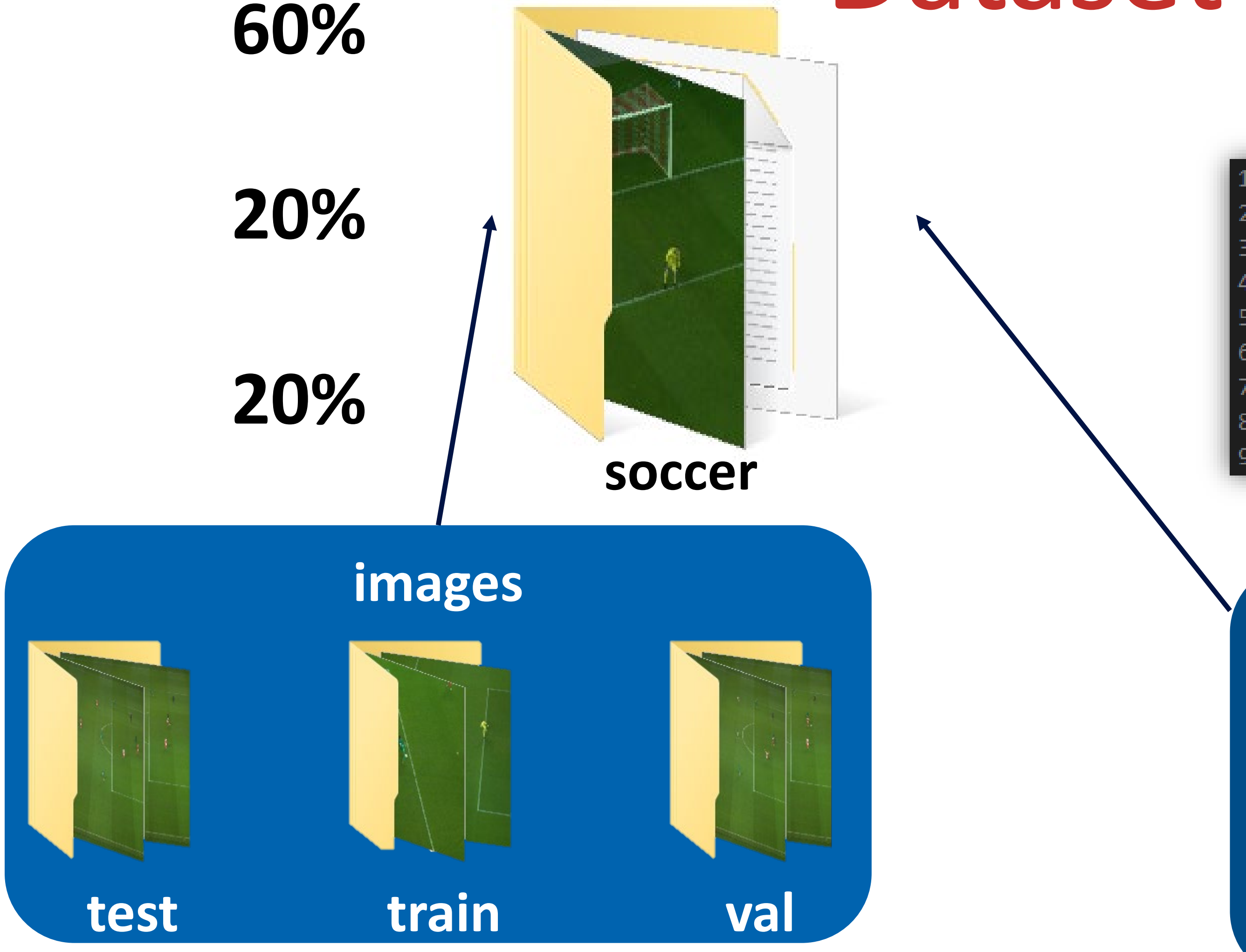


Dataset

60%

20%

20%




```yaml
path: "./soccer" # Path relative to the `train.py` script.
train: images/train
val: images/val

# Classes
nc: 3
names: [
    "Unknown", "Ball", "Person"
]
```

soccer.yaml

```
2 0.9811053240740739 0.6527777777777778 0.013136574074074099
2 0.9525752314814814 0.2431069958847737 0.011631944444444391
2 0.746412037037037 0.36342592592592593 0.016203703703703703
2 0.5156828703703703 0.1868312757201646 0.011458333333333307
2 0.4256655092592592 0.3338477366255144 0.012210648148148161
2 0.8054890046296296 0.25669238683127577 0.01865162037037038
2 0.8894386574074074 0.3833847736625515 0.022048611111111057
2 0.8370949074074073 0.9351851851851852 0.02372685185185185 0.07407407407407408
1 0.8007436342592592 0.3405658436213992 0.006047453703703729 0.010308641975308624
2 0.20578703703703705 0.9281378600823046 0.0261574074074074 0.07026748971193411
2 0.8151331018518517 0.38971193415637867 0.017303240740740793 0.05596707818930039
2 0.7034722222222222 0.19444444444444445 0.020138888888888862 0.05349794238683128
2 0.5539351851851851 0.580761316872428 0.015277777777777763 0.0668724279835391
2 0.11574074074074073 0.6481481481481483 0.020833333333333332 0.0617283950617284
2 0.9890624999999998 0.7520576131687243 0.015162037037037062 0.0617283950617284
2 0.21493055555555554 0.1536522633744856 0.009722222222222228 0.051543209876543226
2 0.281886574074074 0.13549382716049382 0.010648148148148134 0.045267489711934145
2 0.5388570601851852 0.36029320987654323 0.016140046296296288 0.05966049382716051
2 0.6171875 0.14969135802469138 0.012152777777777776 0.05246913580246914
2 0.32132523148148145 0.4746399176954733 0.015335648148148147 0.06244855967078189
2 0.66796875 0.4174382716049383 0.01417824074074074 0.06018518518518519
2 0.1720486111111111 0.2559670781893004 0.013078703703703716 0.057201646090535005
2 0.8214699074074073 0.3669753086419753 0.012152777777777776 0.05617283950617287
2 0.8574652777777776 0.34969135802469137 0.035416666666666693 0.02880658436213992
2 0.004918981481481481 0.38991769547325106 0.009837962962962962 0.05349794238683128
2 0.8006944444444444 0.3476851851851852 0.016550925925925872 0.059567901234567885
2 0.12861689814814814 0.2835905349794239 0.011863425925925925 0.054835390946502076
2 0.7350405092592592 0.31790123456790126 0.015335648148148147 0.05555555555555556
2 0.11168981481481481 0.014300411522633744 0.009259259259259259 0.027777777777777778
```

Demo – Fine Tuning with Azure ML

Support for the end-to-end Machine Learning lifecycle

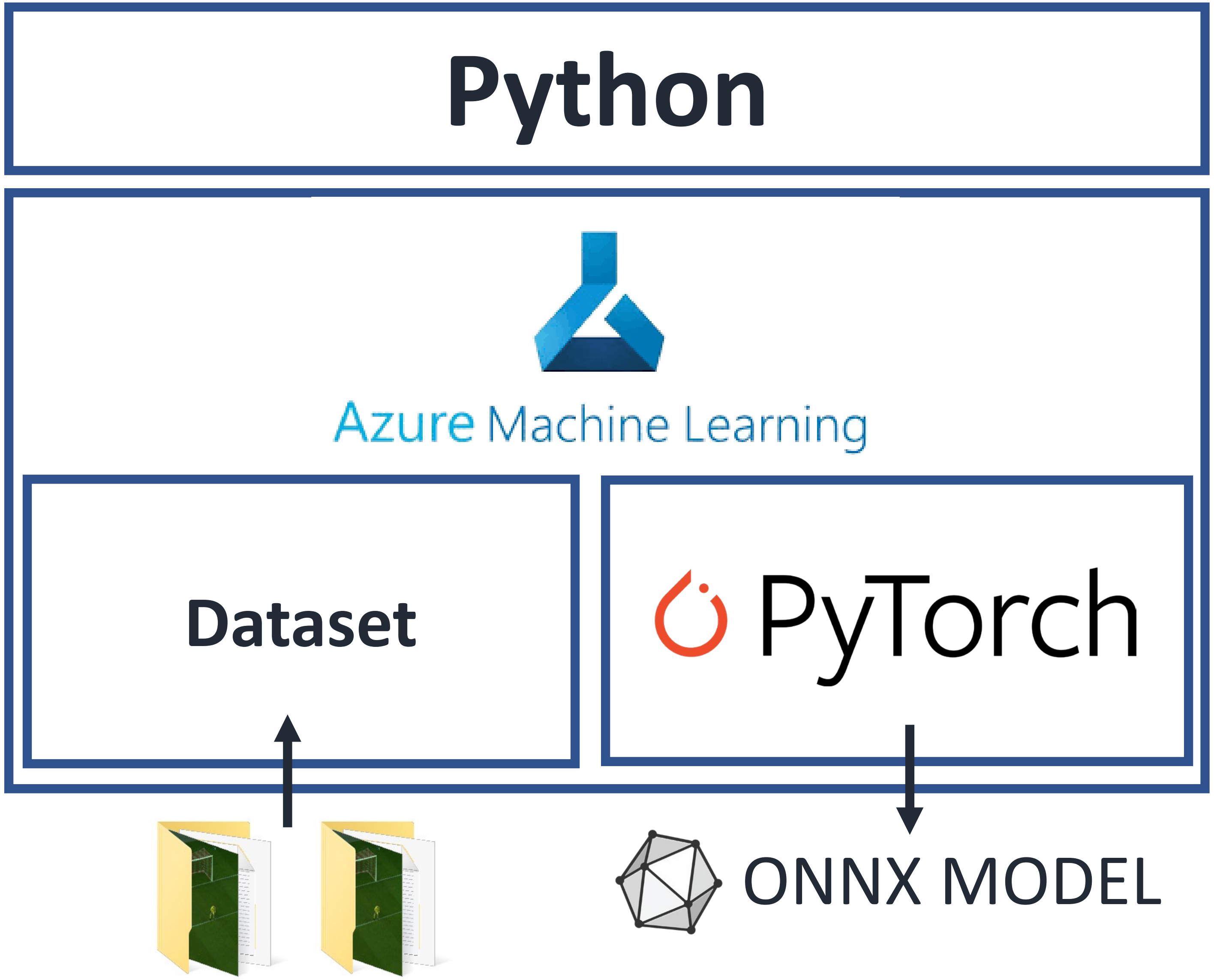
Prepare Data

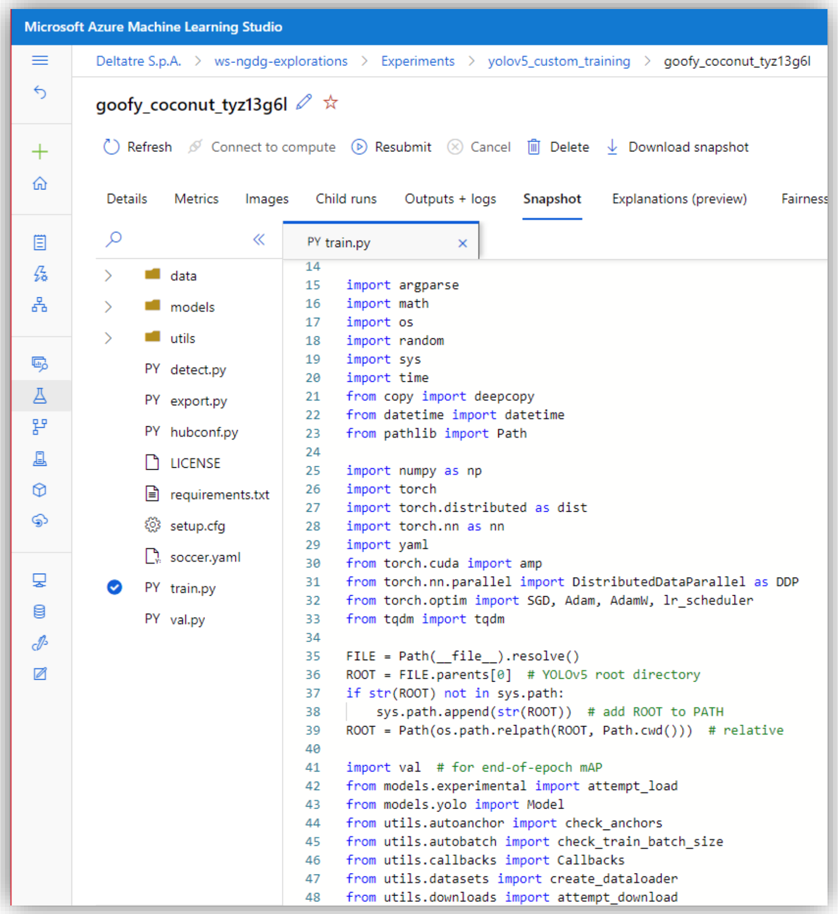
Build and train models

Validate and deploy

Manage and monitor

https://azure.microsoft.com/en-us/services/machine-learning/


Python
Azure Machine Learning
Dataset
PyTorch
ONNX MODEL


Microsoft Azure Machine Learning Studio
Deltatre S.p.A. > ws-ngdg-explorations > Experiments > yolov5_custom_training > goofy_coconut_tyz13g6l
goofy_coconut_tyz13g6l
Refresh
Connect to compute
Resubmit
Cancel
Delete
Download snapshot
Details
Metrics
Images
Child runs
Outputs + logs
Snapshot
Explanations (preview)
Fairness
train.py
data
models
utils
detect.py
export.py
hubconf.py
LICENSE
requirements.txt
setup.cfg
soccer.yaml
train.py
val.py
import argparse
import math
import os
import random
import sys
import time
from copy import deepcopy
from datetime import datetime
from pathlib import Path
import numpy as np
import torch
import torch.distributed as dist
import torch.nn as nn
import yaml
from torch.cuda import amp
from torch.nn.parallel import DistributedDataParallel as DDP
from torch.optim import SGD, Adam, AdamW, lr_scheduler
from tqdm import tqdm
FILE = Path(__file__).resolve()
ROOT = FILE.parents[0] # YOLOv5 root directory
if str(ROOT) not in sys.path:
sys.path.append(str(ROOT)) # add ROOT to PATH
ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative
import val # for end-of-epoch mAP
from models.experimental import attempt_load
from models.yolo import Model
from utils.autoanchor import check_anchors
from utils.autobatch import check_train_batch_size
from utils.callbacks import Callbacks
from utils.datasets import create_dataloader
from utils.downloads import attempt_download

Thank You!

ευχαριστώ Salamat Po متشکرم شكراً Grazie

благодаря ありがとうございます Kiitos Teşekkürler 谢谢

ขอบคุณครับ Obrigado شکریہ Terima Kasih Dziękuję

Hvala Köszönöm Tak Dank u wel дякую Tack

Mulţumesc спасибо Danke Cảm ơn Gracias

多謝晒 Ďakujem תודה நன்றி Děkuji 감사합니다

End-to-End Object Detection with Azure ML & ONNX Runtime

Slides/Demo repository

Deltatre Innovation Lab

GitHub

DOWNLOAD ME

https://github.com/deltatrelabs/deltatre-wpc-2022-demo

Useful links (1/2)

- **https://github.com/deltatrelabs/deltatre-wpc-2022-demo**
- https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/image-classification
- https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/object-detection-onnx
- https://github.com/WongKinYiu/yolov7
- https://github.com/ultralytics/yolov5/releases
- https://github.com/lutzroeder/netron
- https://github.com/daquexian/onnx-simplifier
- https://pytorch.org/
- https://www.tensorflow.org/
- https://github.com/dotnet/TorchSharp
- https://github.com/SciSharp/TensorFlow.NET
- https://docs.microsoft.com/en-us/dotnet/machine-learning/

Useful links (2/2)

- https://azure.microsoft.com/en-us/services/machine-learning/
- https://cvat.org
- https://makesense.ai
- https://labelbox.com
- https://roboflow.com
- https://avaloniaui.net/

About us

Clemente GIORIO @tinux80

R&D Senior Software Engineer @ deltatre

- Augmented/Mixed/Virtual Reality
- Artificial Intelligence, Machine Learning, Deep Learning
- Internet of Things
- Hybrid Clusters
- Multimodal Tracking

MVP Microsoft Most Valuable Professional

dotNET{podcast}

intel Software INNOVATOR

[PACKT] PUBLISHING Author

FAB LAB NAPOLI

About us

Ing. Gianni ROSA GALLINA @giannirg

R&D Technical Lead @ deltatre

- AI, Machine Learning, Deep Learning on multimedia content
- Virtual/Augmented/Mixed Reality
- Immersive video streaming & 3D graphics for sport events
- Cloud solutions, web backends, serverless, video workflows
- Mobile apps dev (Windows / Android / Xamarin)
- End-to-end solutions with Microsoft Azure

MVP Microsoft Most Valuable Professional

Microsoft Specialist
Programming in C#
Programming in HTML5 with JavaScript & CSS3

Microsoft CERTIFIED Solutions Developer
Windows Store Apps Using C#
Web Applications

Microsoft CERTIFIED AZURE AI ENGINEER ASSOCIATE

Microsoft CERTIFIED AZURE DATA SCIENTIST ASSOCIATE

unity Certified Programmer

PLURALSIGHT Author

https://gianni.rosagallina.com/en/

17, 18, 19 OTTOBRE 2022
NH MILANO CONGRESS CENTRE

Powered by OVERNET.

Supported by Microsoft