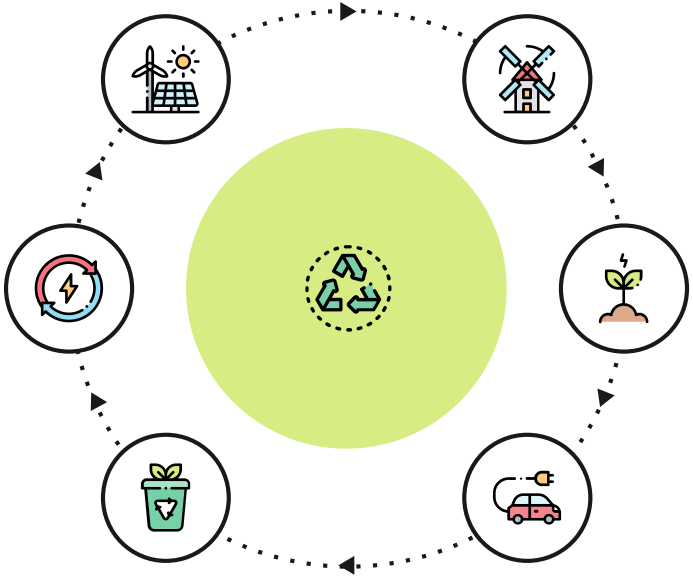
Litter Detection In The Wild Using Efficient Deep Learning Models

Gaviraghi Elia, 869493

Advisor	Bianco Simone
Co-Advisor	Schettini Raimondo



Context

COBOL is a National project funded by the Italian MUR started in December 2023 with the aim of defining an advanced framework for managing the waste disposal process. COBOL involves three universities (the University of Milano - Bicocca, which coordinates the project, the Polytechnic of Milan, and the Gran Sasso Science Institute) and interacts with multiple Italian municipalities interested to experiment with the developed services. COBOL works on the definition of a decentralized data-processing architecture that exploits gamification to involve all the relevant stakeholders in the waste disposal process; model-driven engineering, to enable the creation of highly flexible and executable waste disposal models; computer vision, to help identify, sort and categorize the reported waste effectively; federated learning, to enable the sharing of knowledge among different communities involved in waste management without breaking privacy requirements; and self-adaptation, to ensure the capability to face unexpected events. Early results show that the federated learning architecture can be effectively used to collect reports and computer vision techniques applied to real-life datasets can be used to semi-automate the littering detection process

Initial Analysis

State-of-the-Art

Datasets
TACO, PlastOPol
Wider Face, License PLate

Models
YOLO detector & OVD

Approaches
Object Detection
1-class

Methods

State-of-the-Art

Improve existing methods?
Pursue other approaches?

Tuning NMS

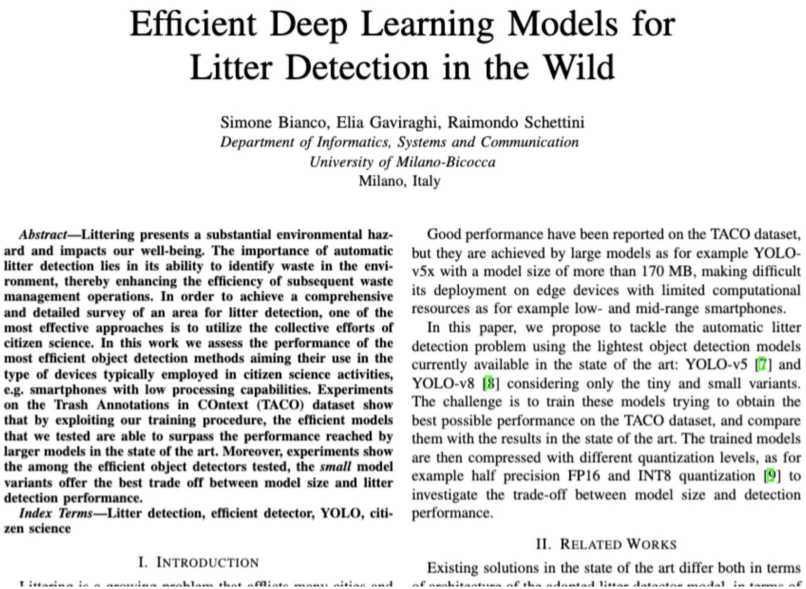
To improve existing research

Two-Stage

To explore new approaches

Contents

Tuning Non-Maximum Suppresion



Efficient Deep Learning Models for Litter Detection in the Wild

Simone Bianco, Elia Gaviraghi, Raimondo Schettini
Department of Informatics, Systems and Communication
University of Milano-Bicocca
Milano, Italy

***Abstract*—Littering presents a substantial environmental hazard and impacts our well-being. The importance of automatic litter detection lies in its ability to identify waste in the environment, thereby enhancing the efficiency of subsequent waste management operations. In order to achieve a comprehensive and detailed survey of an area for litter detection, one of the most effective approaches is to utilize the collective efforts of citizen science. In this work we assess the performance of the most efficient object detection methods aiming their use in the type of devices typically employed in citizen science activities, e.g. smartphones with low processing capabilities. Experiments on the Trash Annotations in COntext (TACO) dataset show that by exploiting our training procedure, the efficient models that we tested are able to surpass the performance reached by larger models in the state of the art. Moreover, experiments show the among the efficient object detectors tested, the *small* model variants offer the best trade off between model size and litter detection performance.**

***Index Terms*—Litter detection, efficient detector, YOLO, citizen science**

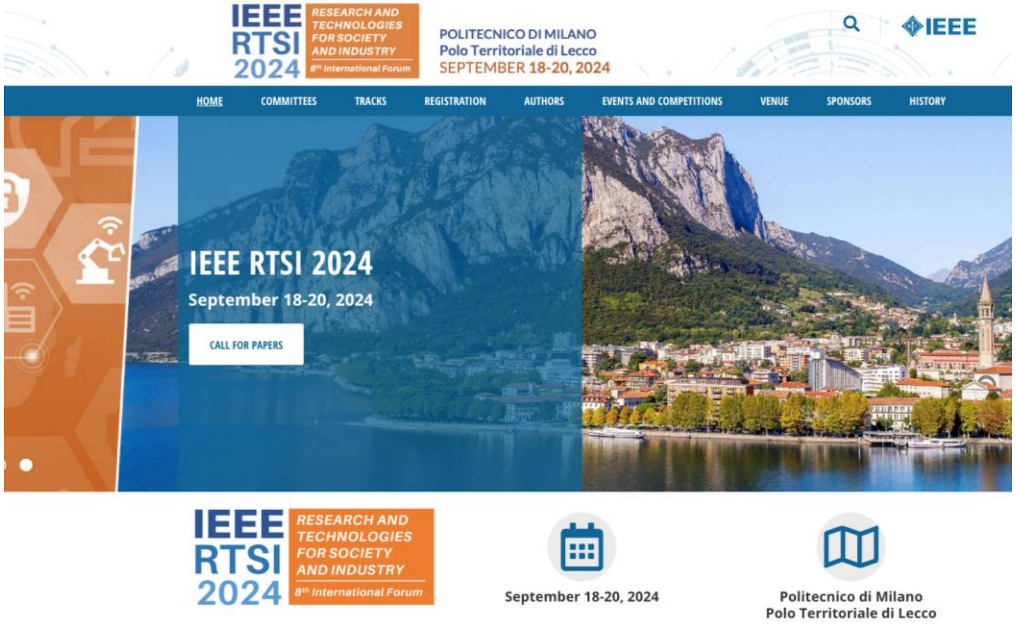
I. Introduction

Good performance have been reported on the TACO dataset, but they are achieved by large models as for example YOLO-v5x with a model size of more than 170 MB, making difficult its deployment on edge devices with limited computational resources as for example low- and mid-range smartphones.

In this paper, we propose to tackle the automatic litter detection problem using the lightest object detection models currently available in the state of the art: YOLO-v5 [7] and YOLO-v8 [8] considering only the tiny and small variants. The challenge is to train these models trying to obtain the best possible performance on the TACO dataset, and compare them with the results in the state of the art. The trained models are then compressed with different quantization levels, as for example half precision FP16 and INT8 quantization [9] to investigate the trade-off between model size and detection performance.

II. Related Works

Existing solutions in the state of the art differ both in terms



Tuning Non-Maximum Suppresion

Predict

YOLO predicted Bounding Boxes

Confidence

Delete Bboxes
Score Conf < Conf threshold

IoU

Delete Bboxes
Intersection Over Union > IoU Thresholds

Tuning Non-Maximum Suppresion

Predict

YOLO predicted Bounding Boxes

Confidence

Delete Bboxes
Score Conf < Conf threshold

IoU

Delete Bboxes
Intersection Over Union > IoU Thresholds

Tuning Non-Maximum Suppresion

Predict

YOLO predicted Bounding Boxes

Confidence

Delete Bboxes
Score Conf < 0.01

IoU

Delete Bboxes
Intersection Over Union > IoU Thresholds

Tuning Non-Maximum Suppresion

Predict

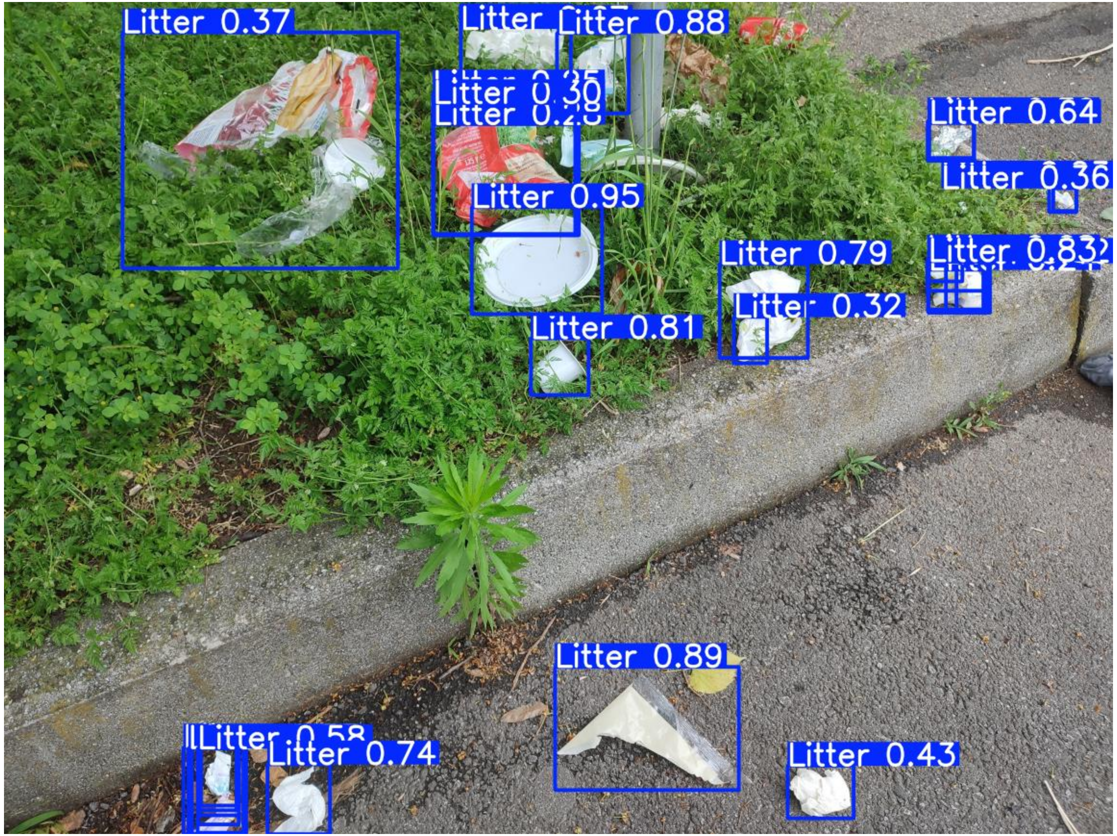
YOLO predicted Bounding Boxes

Confidence

Delete Bboxes
Score Conf < **Tuned**

IoU

Delete Bboxes
Intersection Over Union > IoU Thresholds



Tuning Non-Maximum Suppresion

Predict

YOLO predicted Bounding Boxes

Confidence

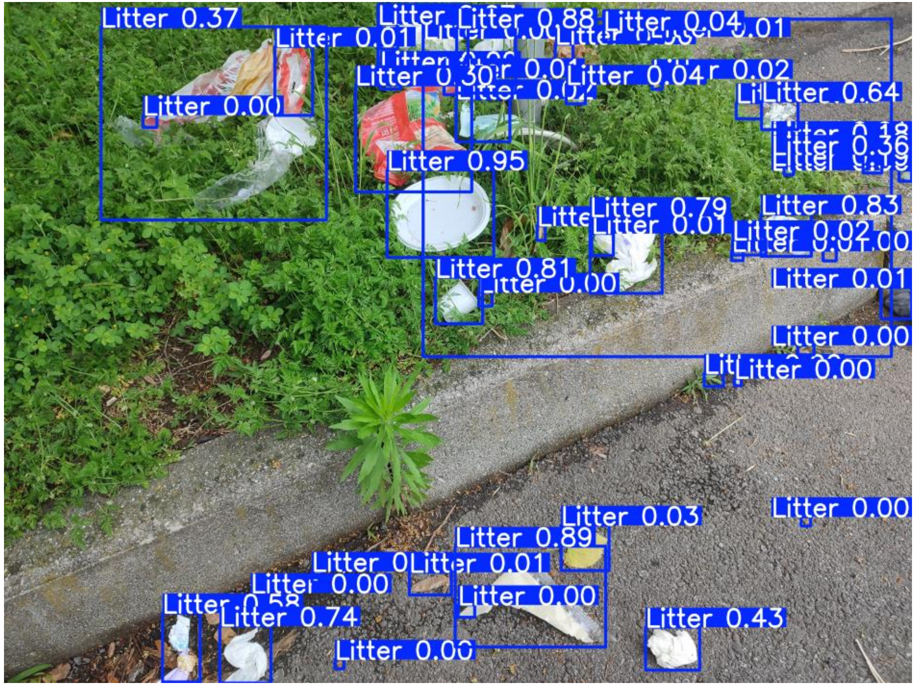
Delete Bboxes
Score Conf < **Tuned**

IoU

Delete Bboxes
Intersection Over Union >
Tuned IoU

Tuning Non-Maximum Suppresion

Default confidence



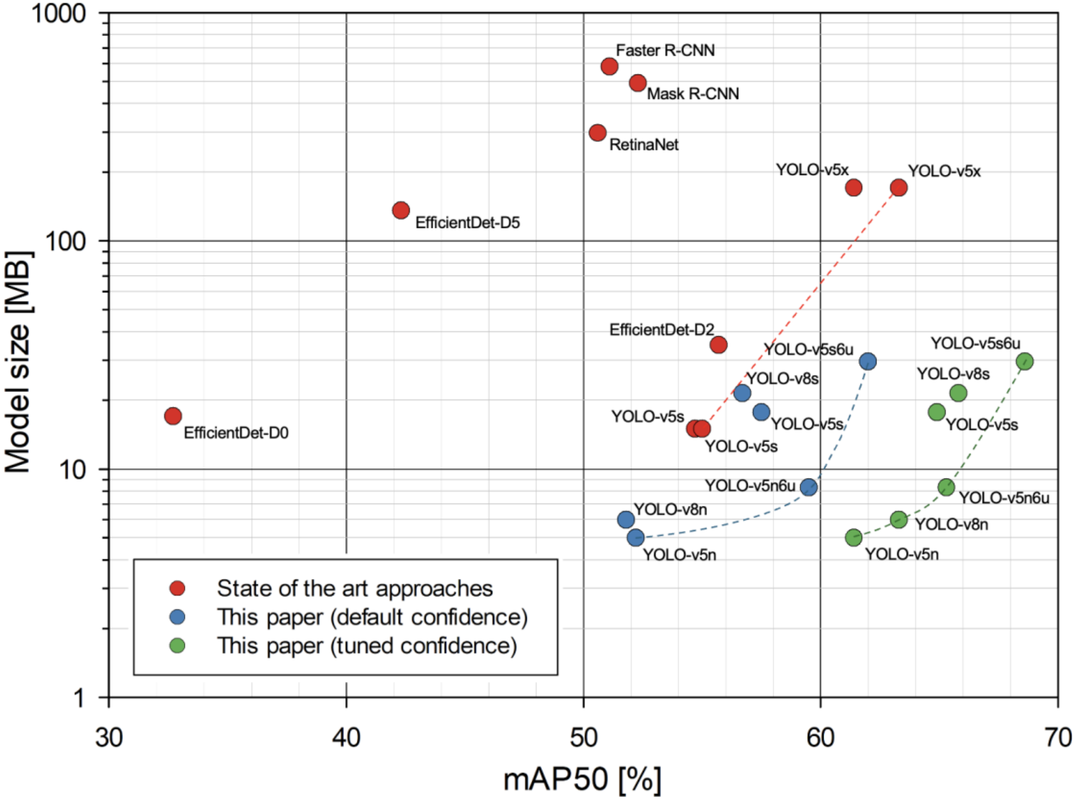
Tuned confidence

NMS tuning Results

Table 5.2: Litter detection results on TACO dataset on the TACO-1 task. The best results are highlighted in bold, and only increments greater than 7.0% are reported.

Method	Default thresholds mAP50	Default thresholds mAP50-95	Default thresholds recall	Default thresholds precision	Tuned thresholds mAP50	Tuned thresholds mAP50-95	Tuned thresholds recall	Tuned thresholds precision	Tuned Value Conf	Tuned Value Iou	Size (MB)
RetinaNet [7]	50.6										297.0
Faster R-CNN [7]	51.1										580.0
Mask R-CNN [7]	52.3										491.0
EfficientDet-D0 [7]	32.7										17.0
EfficientDet-D2 [25]	56.8	40.4									35.0
EfficientDet-D5 [7]	42.3										136.0
YOLO-v5s [7]	54.7										15.0
YOLO-v5s [8]	55.0	38.5									15.0†
YOLO-v5x [7]	**63.3**										171.0
YOLO-v5x [8]	61.4	**47.6**									171.0†
YOLO-v5n (this work)	52.2	34.3	47.2	69.2	$62.1^{+\mathbf{9.9\%}}$	$44.5^{+10.2\%}$	46.8	73.4	0.25	0.2	**5.0**
YOLO-v5n6u (this work)	59.5	41.7	50.9	73.8	$66.7^{+7.2\%}$	$\mathbf{50.5}^{+8.8\%}$	52.9	73.3	0.25	0.4	8.3
YOLO-v8n (this work)	51.8	34.7	46.6	68.8	$61.6^{+9.8\%}$	$45.4^{+10.7\%}$	46.7	70.2	0.2	0.2	6.0
YOLO-v5s (this work)	57.8	39.3	50.7	77.3	$65.4^{+7.6\%}$	$48.8^{+9.5\%}$	50.3	**77.7**	0.2	0.5	17.7
YOLO-v5s6u (this work)	62.0	40.7	52.7	77.1	**68.1**	$47.9^{+7.2\%}$	**53.9**	76.1	0.3	0.5	29.6
YOLO-v8s (this work)	56.5	39.6	49.3	74.9	$65.2^{+8.7\%}$	$50.2^{+\mathbf{10.6\%}}$	49.4	75.4	0.3	0.6	21.5

NMS tuning Results



Testing TACO models

PlastOPol

Method	Default thresholds mAP50	mAP50-95	recall	precision	Tuned thresholds mAP50	mAP50-95	recall	precision	Tuned Value Conf	Iou	Size (MB)
RetinaNet [7]	73.3	47.2									297.0
Faster R-CNN [7]	75.3	49.6									580.0
Mask R-CNN [7]	73.3	50.2									491.0
EfficientDet-D0 [7]	65.0	51.1									17.0
EfficientDet-D5 [7]	73.2	69.1									136.0
YOLO-v5s [7]	79.9	62.4									15.0
YOLO-v5x [7]	**84.9**	71.1									171.0
YOLO-v5n (this work)	66.6	49.1	57.7	78.6	70.5	56.3$^{+7.2\%}$	60.5	74.3	0.4	0.2	**5.0**
YOLO-v5n6u (this work)	67.9	51.5	61.5	77.0	70.5	57.2	61.4	**77.7**	0.4	0.5	8.3
YOLO-v8n (this work)	64.5	48.6	58.3	75.5	69.7	56.2$^{+7.6\%}$	57.7	76.8	0.3	0.3	6.0
YOLO-v5s (this work)	64.3	49.3	58.1	76.4	69.2	56.6$^{+7.3\%}$	58.1	76.4	0.4	0.5	17.7
YOLO-v5s6u (this work)	66.1	47.4	61.4	73.4	**71.0**	53.8	**65.2**	70.7	0.4	0.4	29.6
YOLO-v8s (this work)	64.9	50.3	59.3	76.1	70.6	**58.8**$^{+8.5\%}$	59.2	76.7	0.55	0.5	21.5

UniMiB

Method	Default thresholds mAP50	mAP50-95	recall	precision	Tuned thresholds mAP50	mAP50-95	recall	precision	Tuned Value Conf	Iou	Size (MB)
YOLO-v5n (this work)	56.0	37.0	48.2	67.4	63.4$^{+7.4\%}$	46.6$^{+9.6\%}$	44.7	**78.4**$^{+11.0\%}$	0.25	0.2	**5.0**
YOLO-v5n6u (this work)	63.4	45.1	54.6	72.2	67.9	**52.2**$^{+7.1\%}$	53.5	77.0	0.25	0.4	8.3
YOLO-v8n (this work)	54.4	36.8	46.7	67.6	62.6$^{+8.2\%}$	47.0$^{+10.2\%}$	45.0	75.3$^{+7.7\%}$	0.2	0.2	6.0
YOLO-v5s (this work)	57.9	40.3	48.9	73.6	64.9$^{+7.0\%}$	49.2$^{+8.9\%}$	49.7	74.1	0.2	0.5	17.7
YOLO-v5s6u (this work)	63.6	40.9	55.5	73.3	**68.5**	47.0	**56.3**	73.9	0.3	0.5	29.6
YOLO-v8s (this work)	58.6	41.1	50.0	72.4	65.3	50.8	48.2	76.8	0.3	0.6	21.5

Contents

YOLO World

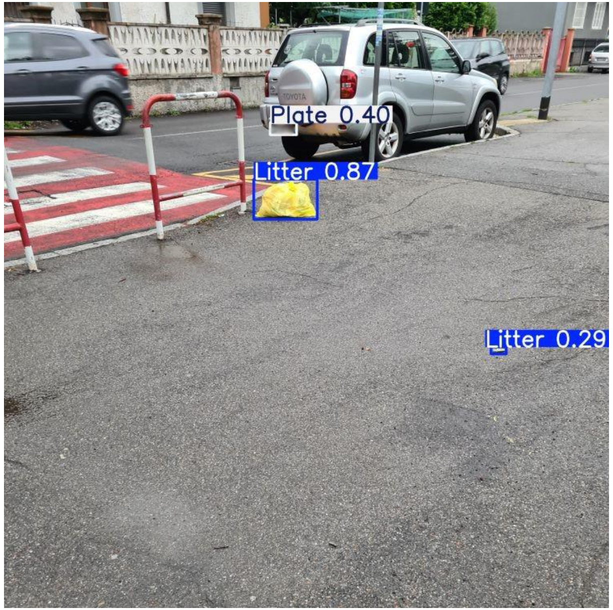
Censoring sensitive information

COBOL is a National project funded by the Italian MUR started in December 2023 with the aim of defining an advanced framework for managing the waste disposal process. COBOL involves three universities (the University of Milano - Bicocca, which coordinates the project, the Polytechnic of Milan, and the Gran Sasso Science Institute) and interacts with multiple Italian municipalities interested to experiment with the developed services. COBOL works on the definition of a decentralized data-processing architecture that exploits gamification to involve all the relevant stakeholders in the waste disposal process; model-driven engineering, to enable the creation of highly flexible and executable waste disposal models; computer vision, to help identify, sort and categorize the reported waste effectively; ***federated learning, to enable the sharing of knowledge among different communities involved in waste management without breaking privacy requirements;*** *and self-adaptation, to ensure the capability to face unexpected events. Early results show that the federated learning architecture can be effectively used to collect reports and computer vision techniques applied to real-life datasets can be used to semi-automate the littering detection process*

No need to share images with a central authority

YOLO World

Original	YOLO v8s	Yolo World X

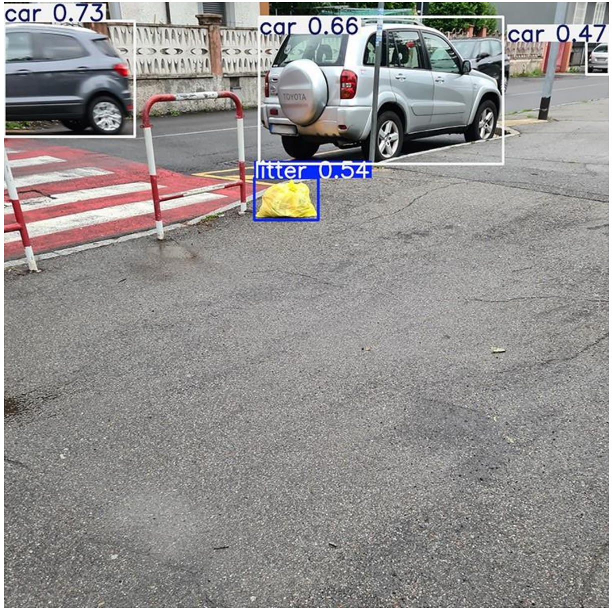


YOLO World - Wider Face

Method	Set	Default thresholds mAP50	Default thresholds mAP50-95	Tuned thresholds mAP50	Tuned thresholds mAP50-95	Tuned Value Conf	Tuned Value Iou
ACF [41]	Easy	69.5					
Faceness [41]	Easy	71.3					
Multiscale Cascade CNN [41]	Easy	71.1					
Two-Stage CNN [41]	Easy	65.7					
YOLO-v8n (this work)	Easy	75.7	44.1	81.2	$52.7^{8.6\%}$	0.2	0.2
YOLO-v8s (this work)	Easy	79.4	47.2	**83.6**	$\mathbf{54.6}^{7.4\%}$	0.2	0.5
YOLO World S zero shot (1 class)	Easy	22.6	9.5	$35.2^{12.6\%}$	14.6	def	0.01
[illegible]	Easy	[illegible]	[illegible]			def	def
YOLO World X zero shot (1 class)	Easy	33.9	17.0	$48.3^{14.4\%}$	$24.5^{7.5\%}$	def	0.01
[illegible]	Easy	[illegible]	[illegible]			def	def
ACF [41]	Medium	58.8					
Faceness [41]	Medium	60.4					
Multiscale Cascade CNN [41]	Medium	63.6					
Two-Stage CNN [41]	Medium	58.9					
YOLO-v8n (this work)	Medium	61.5	34.2	$72.0^{10.5\%}$	$45.1^{10.9\%}$	0.2	0.2
YOLO-v8s (this work)	Medium	66.6	37.7	$\mathbf{74.6}^{8.0\%}$	$\mathbf{47.0}^{9.3\%}$	0.2	0.5
YOLO World S zero shot (1 class)	Medium	17.4	6.9	$30.3^{12.9\%}$	11.5	def	0.01
[illegible]	Medium	[illegible]	[illegible]			def	def
YOLO World X zero shot (1 class)	Medium	27.7	13.2	$42.9^{15.2\%}$	$20.7^{7.5\%}$	def	0.01
[illegible]	Medium	[illegible]	[illegible]			def	def

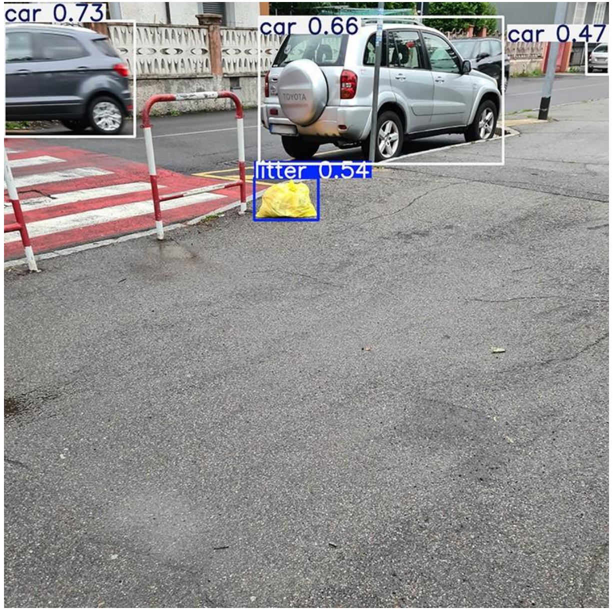
YOLO World Two-Stage

YOLO World X – first prediction

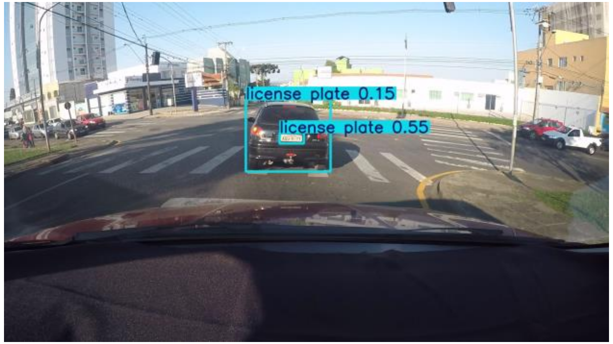


Predictions on crops

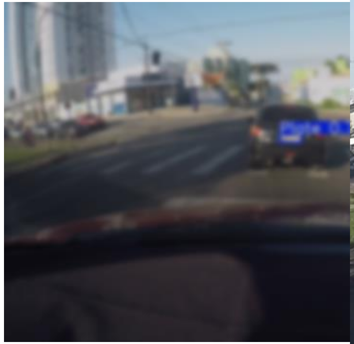
YOLO World Two-Stage

Original	YOLO World X – first prediction	YOLO World - final
		

YOLO World Two-Stage

YOLO v5n6u	YOLO World X	YOLO World - final
		

YOLO World Two-Stage



YOLO World Two-Stage, Wider Face

Method	Set	Default thresholds		Tuned thresholds		Tuned Value	
		mAP50	mAP50-95	mAP50	mAP50-95	Conf	Iou
ACF [41]	Easy	69.5					
Faceness [41]	Easy	71.3					
Multiscale Cascade CNN [41]	Easy	71.1					
Two-Stage CNN [41]	Easy	65.7					
YOLO-v8n (this work)	Easy	75.7	44.1	81.2	$52.7^{8.6\%}$	0.2	0.2
YOLO-v8s (this work)	Easy	79.4	47.2	**83.6**	$\mathbf{54.6}^{7.4\%}$	0.2	0.5
YOLO World S zero shot (1 class)	Easy	22.6	9.5	$35.2^{12.6\%}$	14.6	def	0.01
YOLO World S zero shot (two stage)	Easy	61.8	31.7			def	def
YOLO World X zero shot (1 class)	Easy	33.9	17.0	$48.3^{14.4\%}$	$24.5^{7.5\%}$	def	0.01
YOLO World X zero shot (two stage)	Easy	66.0	39.9			def	def
ACF [41]	Medium	58.8					
Faceness [41]	Medium	60.4					
Multiscale Cascade CNN [41]	Medium	63.6					
Two-Stage CNN [41]	Medium	58.9					
YOLO-v8n (this work)	Medium	61.5	34.2	$72.0^{10.5\%}$	$45.1^{10.9\%}$	0.2	0.2
YOLO-v8s (this work)	Medium	66.6	37.7	$\mathbf{74.6}^{8.0\%}$	$\mathbf{47.0}^{9.3\%}$	0.2	0.5
YOLO World S zero shot (1 class)	Medium	17.4	6.9	$30.3^{12.9\%}$	11.5	def	0.01
YOLO World S zero shot (two stage)	Medium	52.8	26.8			def	def
YOLO World X zero shot (1 class)	Medium	27.7	13.2	$42.9^{15.2\%}$	$20.7^{7.5\%}$	def	0.01
YOLO World X zero shot (two stage)	Medium	63.8	37.2			def	def

YOLO World Two-Stage

Method	Default thresholds				Tuned thresholds				Tuned Value	
	mAP50	mAP50-95	recall	precision	mAP50	mAP50-95	recall	precision	Conf	Iou
YOLO-v5n (this work)	84.3	45.5	80.7	85.0	86.4	50.8	83.3	85.9	0.2	0.4
YOLO-v5n6u (this work)	86.1	45.6	83.6	82.9	88.5	49.9	85.7	85.5	0.1	0.4
YOLO-v8n (this work)	88.4	52.7	85.7	89.2	90.0	**57.2**	85.7	89.7	0.2	0.5
YOLO-v5s (this work)	88.3	52.1	85.8	84.7	89.9	56.9	**88.6**	84.2	0.1	0.4
YOLO-v5s6u (this work)	71.7	33.1	68.7	76.2	76.7	38.1	74.4	76.9	0.2	0.2
YOLO-v8s (this work)	87.3	48.5	83.9	85.2	89.2	53.4	87.9	85.5	0.1	0.3
YOLO World S zero shot (1 class)	40.5	16.5	43.5	48.1	$47.9^{7.2\%}$	19.5	58.3	46.3	def	0.01
YOLO World X zero shot (1 class)	47.4	21.2	63.1	47.8	$57.0^{9.6}$	25.6	64.9	50.4	def	0.01
YOLO World S zero shot (4 classes)	43.2	17.0	66.7	25.6	45.0	16.8	56.5	40.2	def	0.01
YOLO World X zero shot (4 classes)	60.1	26.8	68.9	50.6	61.7	27.1	74.9	55.7	def	0.01
YOLO World S zero shot (two stage)	**90.3**	56.0							def	def
YOLO World X zero shot (two stage)	87.3	52.8							def	def

License Plates

Contents

UniMiB Trash Dataset

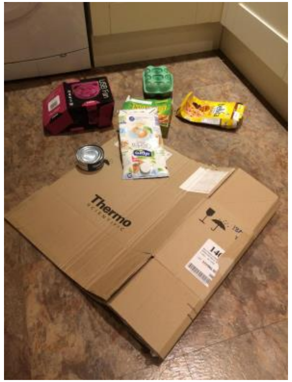
Why do we need to collect a new dataset?

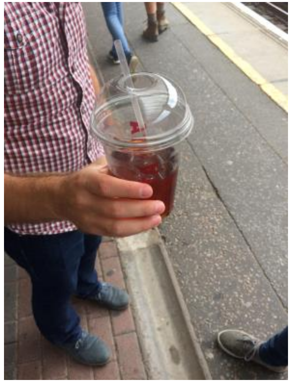
- The **number of images** in the datasets considered is very low for deep learning models
  - TACO: 1.500 annotated images
  - PlastOPol: 2.418 annotated images
- One class or unrealistic classes
  - PlastOPol: only 1 class
  - TACO: 60 classes, high class imbalance, poor representation
- Existing datasets are not always realistic
  - Some images do not contain real abandoned waste.
- Prevalence of images with small waste scenarios
  - TACO: 4784 annotations, ~3.19 for images
  - PlastOPol: 5.300 annotations, ~2.2 for images

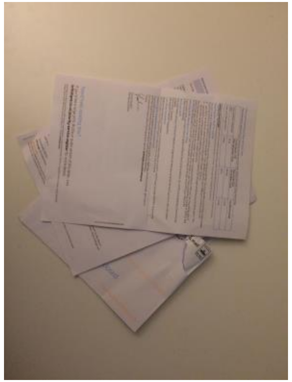
Imaging and Vision

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Litter in the Wild?







Complex Scenarios

TACO typical scenarios

More complex scenarios

Cluster Problem

Results on UniMiB Trash

Method	Class	Default thresholds mAP50	mAP50-95	recall	precision	Tuned thresholds mAP50	mAP50-95	recall	precision	Tuned Value Conf	Iou	Size (MB)
	average	42.6	31.2	43.9	51.8	44.5	33.8	43.1	53.2			
	large	44.1	30.2	54.5	47.8	44.2	30.2	50.0	51.1			
YOLO-v5n (this work)	medium	36.0	28.5	42.9	47.9	39.3	32.1	43.5	49.8	0.1	0.4	**5.0**
	small	69.4	54.6	66.7	68.8	71.9	58.7	66.3	70.2			
	micro	20.8	11.7	11.3	42.8	22.7	14.2	12.5	41.8			
	average	53.3	42.0	49.9	58.7	55.9	**45.7**	**58.9**	53.7			
	large	48.1	36.7	50.0	48.9	54.5	42.8	63.6	49.2			
YOLO-v5n6u (this work)	medium	37.9	31.7	43.8	43.5	39.6	34.8	49.3	43.0	0.2	0.5	8.3
	small	77.4	65.0	70.1	**76.4**	**78.4**	67.9	**75.9**	69.0			
	micro	49.9	34.5	35.8	**65.9**	51.3	37.4	46.7	53.5			
	average	44.4	33.0	47.2	51.0	46.7	36.9	46.7	51.9			
	large	51.3	39.0	59.1	55.2	48.7	38.4	54.5	57.1			
YOLO-v8n (this work)	medium	34.0	25.7	35.6	48.4	41.8	32.7	35.6	48.1	0.2	0.3	6.0
	small	68.7	54.7	69.9	64.4	70.6	59.3	70.1	66.3			
	micro	23.4	12.7	24.2	35.8	28.1	17.2	26.7	35.8			
	average	44.0	32.1	40.6	58.9	45.6	35.3	47.8	47.6			
	large	37.5	24.8	40.9	52.4	40.3	26.5	40.9	54.6			
YOLO-v5s (this work)	medium	38.7	33.4	42.5	35.4	39.0	33.6	42.8	39.0	0.1	0.2	17.7
	small	73.0	61.5	73.1	65.4	73.1	61.6	73.0	66.1			
	micro	33.1	21.6	34.8	37.0	33.2	21.6	35.7	36.5			
	average	54.4	41.0	54.8	58.8	**56.1**	44.9	55.2	**60.7**			
	large	43.2	28.7	54.5	49.7	44.7	31.7	54.5	55.7			
YOLO-v5s6u (this work)	medium	42.4	31.6	45.2	49.9	**45.8**	**37.0**	45.2	**51.3**	0.35	0.5	29.6
	small	78.0	65.9	71.6	74.9	77.1	**68.6**	72.2	75.2			
	micro	54.0	37.8	47.8	60.6	**56.6**	**42.1**	**48.8**	60.5			
	average	44.6	32.4	48.9	45.3	47.4	37.6	46.5	49.3			
	large	52.4	37.1	50.0	59.3	**56.8**	**45.4**	45.5	**66.7**			
YOLO-v8s (this work)	medium	28.1	21.6	38.4	29.9	29.5	25.1	38.4	32.6	0.2	0.6	21.5
	small	69.1	54.6	73.3	56.2	70.2	58.4	71.9	60.2			
	micro	28.8	16.3	33.9	35.7	32.9	21.5	30.1	37.7			
	average	14.8	9.4	48.8	15.0	15.8	10.2	55.7	12.3			
	large	8.3	6.4	**81.8**	2.2	7.5	5.7	72.7	1.5			
YOLO-World X (zero shot)	medium	15.7	12.3	76.7	6.6	15.8	12.3	**80.8**	5.4	0.01	def	140.0
	small	33.8	18.4	34.5	49.6	37.3	21.7	56.9	38.0			
	micro	1.5	0.4	2.3	1.7	2.6	0.9	12.2	4.6			

Contents

Triple Model

Class	Method	Training set	mAP50	mAP50-95	recall	precision	Conf	Iou
Litter	YOLO-v8n	Single	61.6	45.4	46.7	70.2	0.2	0.2
		Triple Light	59.8	42.8	42.2	72.2	0.25	0.4
		Triple Large	60.4	42.6	40.1	75.7	0.2	0.4
		Triple Trash Large	**61.5**$^{-0.1}$	**45.5**$^{+0.1}$	**43.7**$^{-3.0}$	**76.6**$^{+6.4}$	0.2	0.4
	YOLO-v8s	Single	65.2	50.2	49.4	75.4	0.3	0.6
		Triple Light	57.9	42.8	43.8	65.7	0.2	0.5
		Triple Large	61.5	45.3	41.8	**76.4**$^{+1.0}$	0.2	0.3
		Triple Trash Large	**63.0**$^{-2.2}$	**48.0**$^{-2.2}$	**49.2**$^{-0.2}$	68.3	0.2	0.5
Face	YOLO-v8n	Single	75.3	47.9	57.5	87.7	0.2	0.5
		Triple Light	71.2	44.4	51.6	85.3	0.25	0.4
		Triple Large	**74.9**$^{-0.4}$	**47.9**	**56.7**$^{-0.8}$	88.2	0.2	0.4
		Triple Trash Large	74.6	47.6	54.0	**91.3**$^{+3.6}$	0.2	0.4
	YOLO-v8s	Single	77.8	49.7	61.7	88.5	0.2	0.5
		Triple Light	74.2	46.2	55.4	88.3	0.2	0.5
		Triple Large	**77.7**$^{-0.1}$	49.7	60.1	**90.7**$^{+2.2}$	0.2	0.3
		Triple Trash Large	77.5	**49.8**$^{+0.1}$	**61.3**$^{-0.4}$	88.0	0.2	0.5
License Plate	YOLO-v8n	Single	90.0	57.2	85.7	89.7	0.2	0.5
		Triple Light	88.5	57.1	**83.3**$^{+2.4}$	87.5	0.25	0.4
		Triple Large	89.1	58.5	82.7	**88.0**$^{-1.7}$	0.2	0.4
		Triple Trash Large	**89.7**$^{-0.3}$	**58.8**$^{+1.6}$	82.7	86.7	0.2	0.4
	YOLO-v8s	Single	89.2	53.4	87.9	85.5	0.1	0.3
		Triple Light	90.9	**63.3**$^{+9.9}$	87.5	82.6	0.2	0.5
		Triple Large	**92.5**$^{+3.3}$	61.4	**89.3**$^{+1.4}$	**88.8**$^{+3.3}$	0.2	0.3
		Triple Trash Large	91.8	62.7	88.7	86.1	0.25	0.5

One YOLO v8s
it is better than
Three YOLO v8n

Contents

Conclusion

Future Developments

NMS tuning
Training validation

UniMiB Trash
Collect more, more, more, more, more images

TWO-STAGE
Exploitation phase

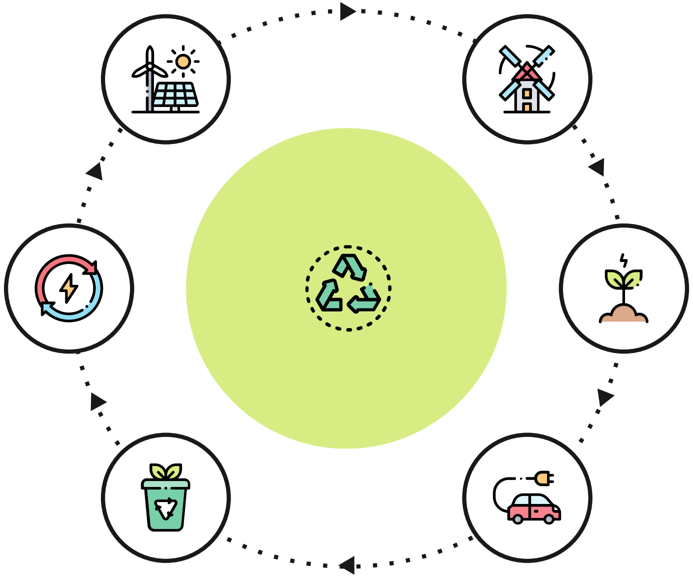
Federated Learning
Using which dataset?

YOLO World
Modify the architecture?

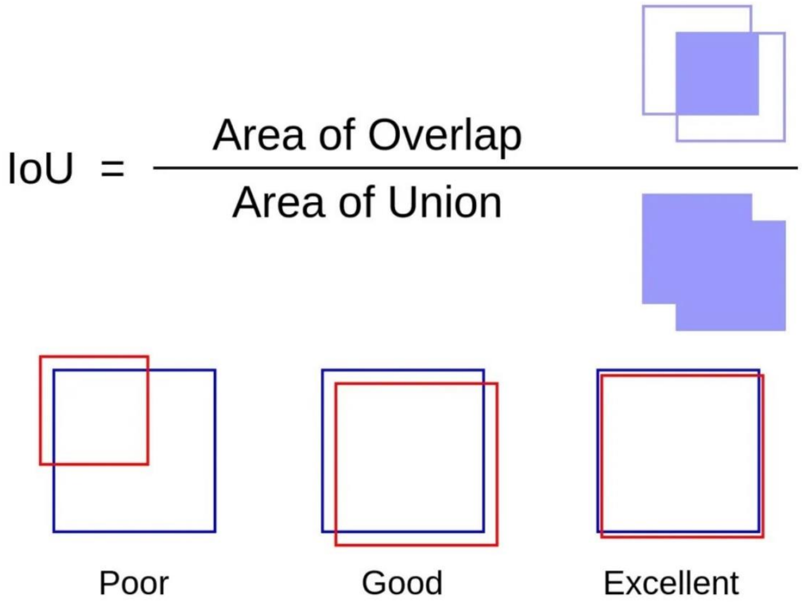
Models
YOLO v10/v11?
RT-DETR?

Supplementary slides

Elia Gaviraghi
Matr 869493

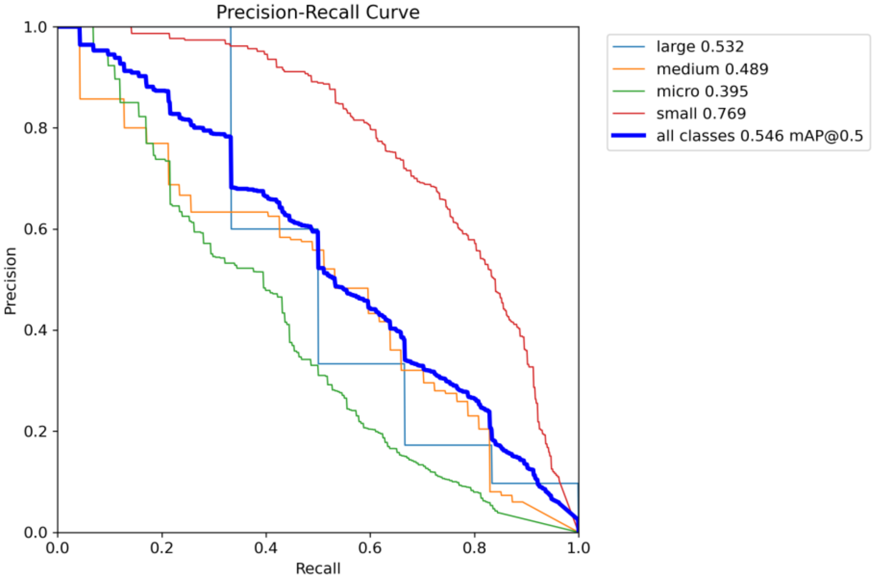


Metrics



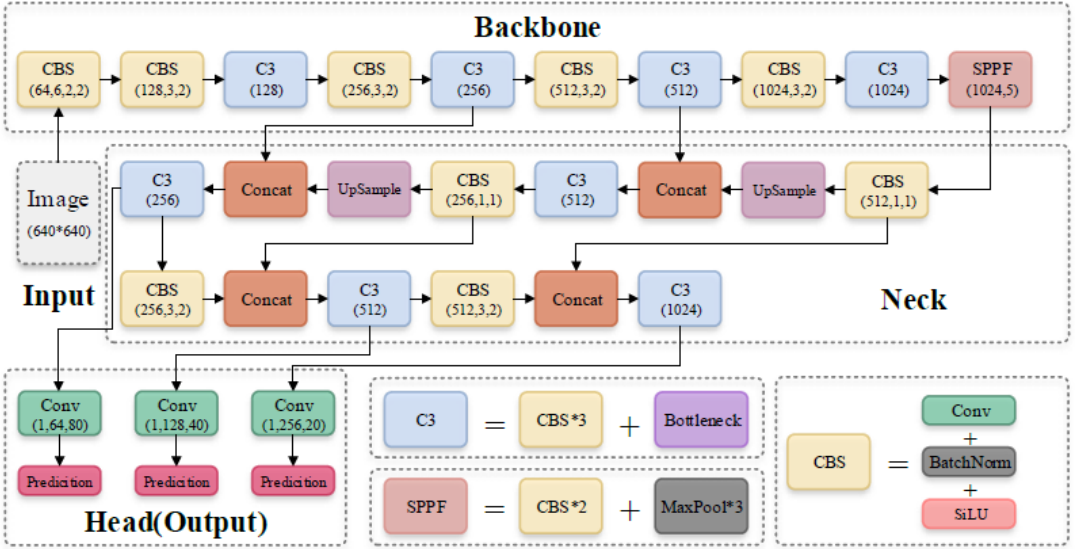
$$\mathrm{AP} = \sum_{n=1}^{N} (R_n - R_{n-1}) \times P_n$$

where P_n is the precision at n-th threshold; R_n is the recall at n-th threshold; N is the number of thresholds.



YOLO Architecture

- **Backbone**
  - A convolutional neural network that aggregates and forms image features
- **Neck**
  - A series of layers to mix and combine feature maps from different stages of the backbone, and pass them forward to prediction
- **Head**
  - responsible for generating the final output.



YOLO World Architecture