A Survey on Latency-Aware/Streaming Perception

Autonomous Vision Seminar

Guray Ozgur
University of Tübingen

Worakorn Ruangratanawicha
University of Tübingen

Agenda

- Introduction
  - Goal and Motivation
  - Prior Work
- Approach
  - Streaming Accuracy
  - Timely Accuracies
  - Dual-Flow Perception and Trend-Aware Loss
  - Learning Informative Metrics with Reinforcement Learning
- Results
  - Qualitative Results
  - Quantitative Results
  - Limitations
- Summary

Introduction

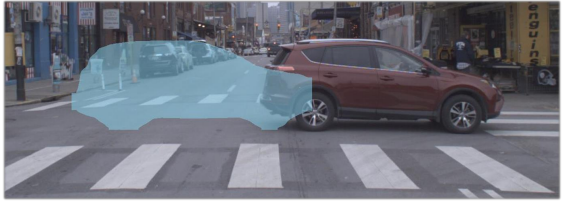
- Goal
- Motivation
- Prior Work

Goal

- **High Accuracy:** Perception systems need to observe their environment as accurate as possible.
- **Low Latency:** Perception systems need to respond quickly in dangerous situations.

Motivation

- **Accuracy-Latency Trade-off:** Accuracy and speed are intertwined with each other.
- **Dynamic World:** By the time an algorithm finishes processing a particular frame, the surrounding world has changed.
- **Offline Evaluation:** Either accuracy or latency is measured in an offline setting.



Guray O., Worakorn R. - A Survey on Latency-Aware/ Streaming Perception, Image Credits: Li: Towards Streaming Perception

Prior Work on 2D Object Detection

Existing domain-specific image object detectors are often classified into two types:

- **Two-stage Detectors:** exceptional localization and object identification accuracy (Fast R-CNN)
- **One-stage Detectors:** fast inference (YOLO, SSD, RetinaNet)
- **Evaluation:**
  - Average Precision (AP)
  - Frames Per Second (FPS)
  - Latency

Performance Evaluation

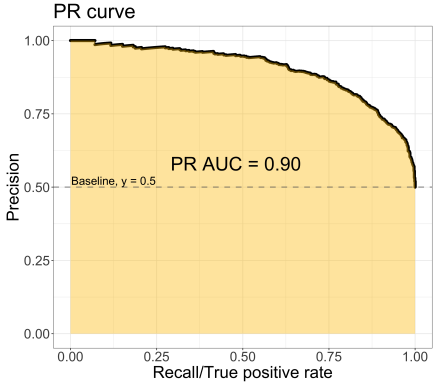
Average Precision Metric:

1. Run detector with varying thresholds
2. Assign detections to closest object
3. Count TP, FP, FN
4. Compute **Average Precision (AP)**

True Positives TP: Number of objects correctly detected (IoU > 0.5)

False Negatives FN: Number of objects not detected (IoU < 0.5)

False Positives FP: Wrong detections

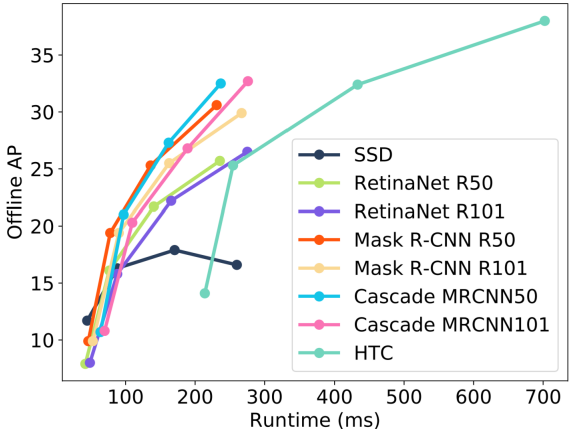


$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detections}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truths}}$$

Pareto optimal latency-accuracy curve

- Prior work routinely explores the trade-off between detection accuracy versus run-time.



In real-time settings, latency-accuracy trade-off is much more important as the agent has to not only **accurately perceive** (detect/track) but also **react** (predict/plan) **as soon as possible**.

Accuracy and latency must be evaluated at the same time.

Approach

- Streaming Accuracy
- Timely Accuracies
- Dual-Flow Perception and Trend-Aware Loss
- Learning Informative Metrics with Reinforcement Learning

Streaming Accuracy

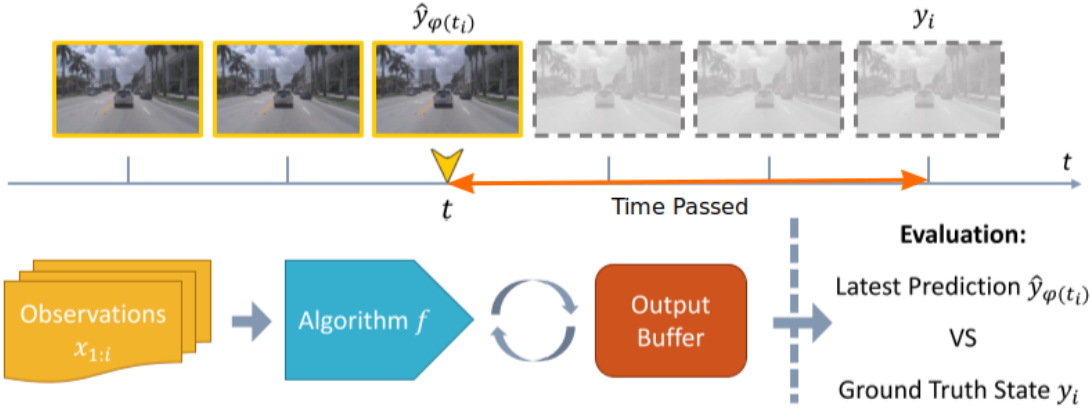
is a new metric to measure the **joint effects** of both **accuracy** and **latency** by putting a real-time constraint to the predictions, the **latest prediction** is compared with the **current ground-truth** at every time instance.

$$\left\{\left(y_i, \hat{y}_{\varphi(t_i)}\right)\right\}_{i=1}^{T}$$

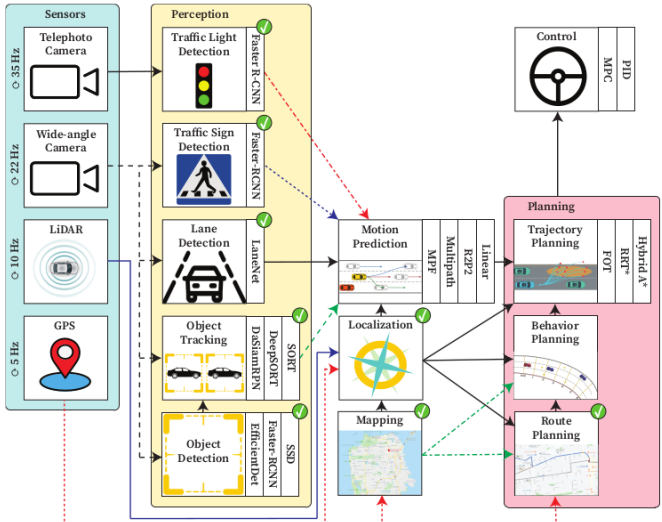
where $\varphi(t) = \arg\max_j s_j < t$ is the real-time constraint. The loss evaluated by the new formed pairs is called streaming loss.

$$L_{\text{streaming}} = L\left(\left\{\left(y_i, \hat{y}_{\varphi(t_i)}\right)\right\}_{i=1}^{T}\right) \tag{1}$$

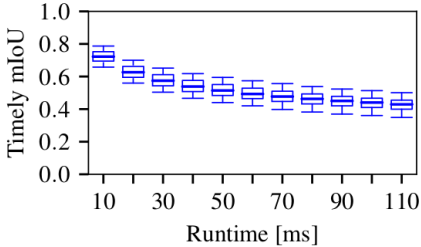
Streaming Accuracy



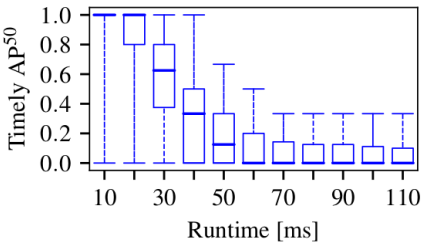
AV Pipeline


Sensors
Telephoto Camera
35 Hz
Wide-angle Camera
22 Hz
LiDAR
10 Hz
GPS
5 Hz
Perception
Traffic Light Detection
Faster R-CNN
Traffic Sign Detection
Faster-RCNN
Lane Detection
LaneNet
Object Tracking
DeepSORT
DaSiamRPN
SORT
Object Detection
EfficientDet
Faster-RCNN
SSD
Motion Prediction
MPF
Multipath
R2P2
Linear
Localization
Mapping
Control
MPC
PID
Planning
Trajectory Planning
FOT
RRT*
Hybrid A*
Behavior Planning
Route Planning

Timely Accuracies



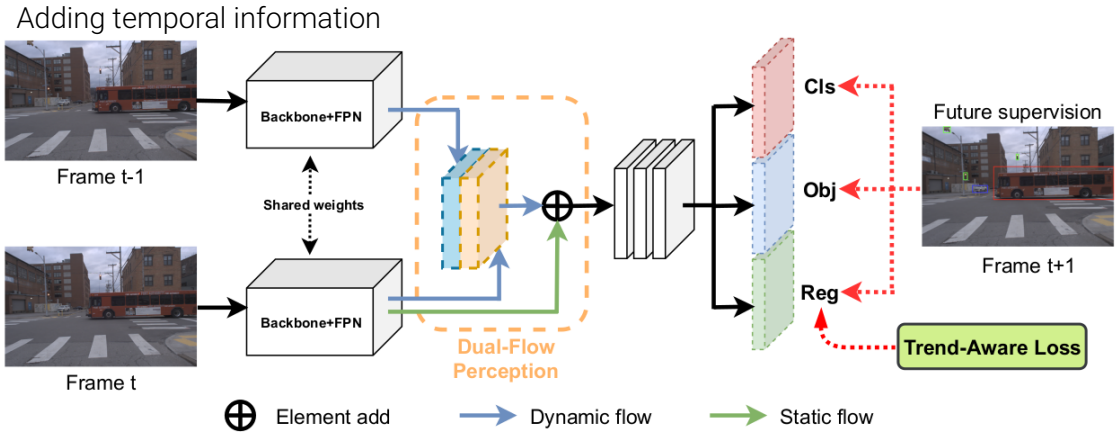
(a) Semantic segmentation timely mIoU.



(b) Pedestrian detection timely AP^{50}.

- Input-output pairs are created considering the run-time of the module.
- In online settings, the accuracy decreases with long run-times.

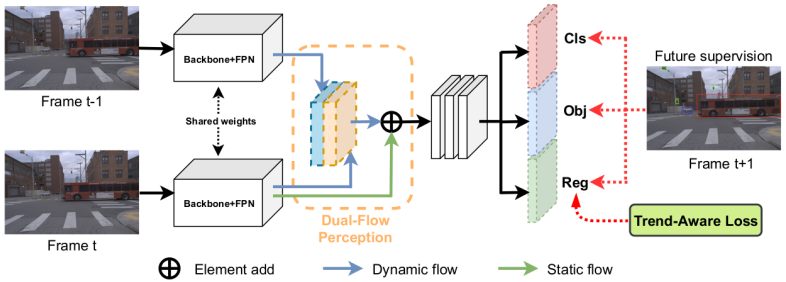
Dual-Flow Perception



The training pipeline for streaming perception using Dual-Flow Perception module (DFP) and Trend-Aware Loss (TAL)

Dual-Flow Perception

- two detectors on two adjacent frames
- aggregating the previous feature map to the current one
- only extract features on the current frame so that the run-time is the same as the base detector



Trend-Aware Loss

- **Matching IoU:** Smaller IoU means fast object.

$$mIoU_i = \max_j \left(\left\{ \mathrm{IoU} \left(box_i^{t+1}, box_j^t \right) \right\} \right) \tag{2}$$

- The trend factor w_i is obtained for each object:

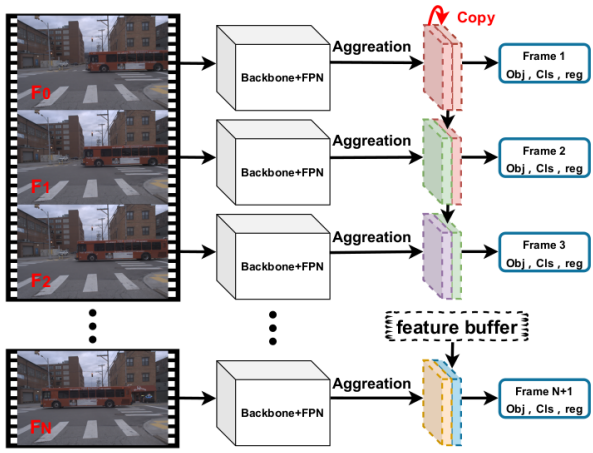
$$\omega_i = \begin{cases} 1/mIoU_i & mIoU_i \geq \tau \\ 1/\nu & mIoU_i < \tau \end{cases} \tag{3}$$

Trend-Aware Loss

By weighting the regression loss with the trend factor $\hat{\omega}_i$, The trend-aware total loss becomes:

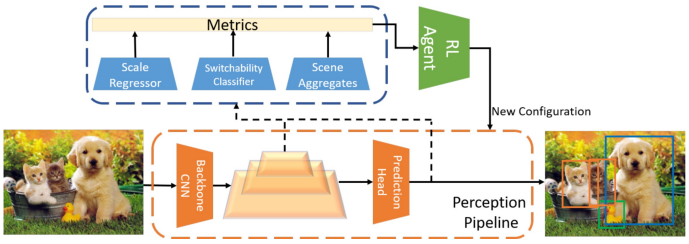
$$\mathcal{L}_{\text{total}} = \sum_{i \in \text{ positive}} \hat{\omega}_i \mathcal{L}_i^{reg} + \mathcal{L}_{cls} + \mathcal{L}_{obj} \tag{4}$$

Inference



The inference pipeline

Learning Informative Metrics with Reinforcement Learning



Using RL to choose 'sweet spot'

learn the metrics for guiding trade-off decisions on run-time.

Informative metrics

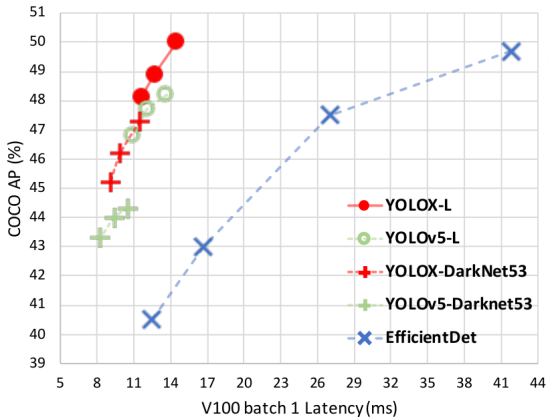
- Switchability
- Adaptive Scale
- Scene Aggregations
  - Confidence agg.
  - Category agg.
  - Object size agg.
  - Crop extents

Informative metrics

- Switchability: (Low/Medium/High)
  It is calculated from standard deviation of IoU across detectors.
- Adaptive Scale:
  The optimal scale has the minimum common foreground object loss.
- Scene Aggregations:
  · Confidence: Average confidence of the prediction
  · Category: Number of instance of each class (Rare objects need more resource.)
  · Size: Areas of detection (Small, Medium, Large)
  · **Crop extents:** It excludes patches with minimal accuracy loss.

Fast Detector for Streaming Perception: **YOLOX**

- superior trade-off between speed and accuracy
- won the first place of Streaming Perception Challenge on WAD 2021



Results

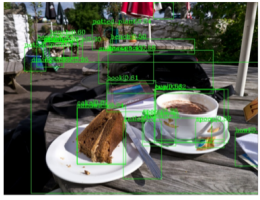
- Qualitative Results
- Quantitative Results
- Limitations

Qualitative Results

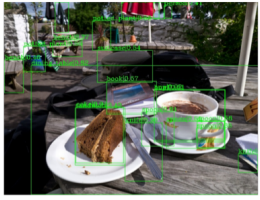
a) Pseudo ground truth b) Real-time latency c) Instance mask forecasting

Instance Segmentation with Streaming Accuracy

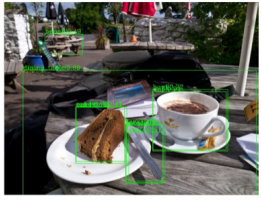
Detection at different input scale



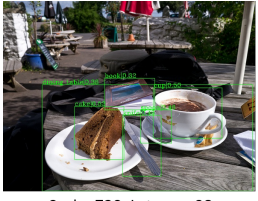
Scale: 720, Latency: 94ms



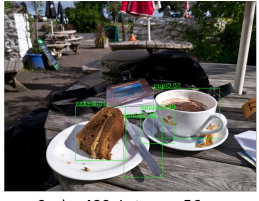
Scale: 480, Latency: 63ms



Scale: 240, Latency: 51ms



Scale: 720, Latency: 98ms

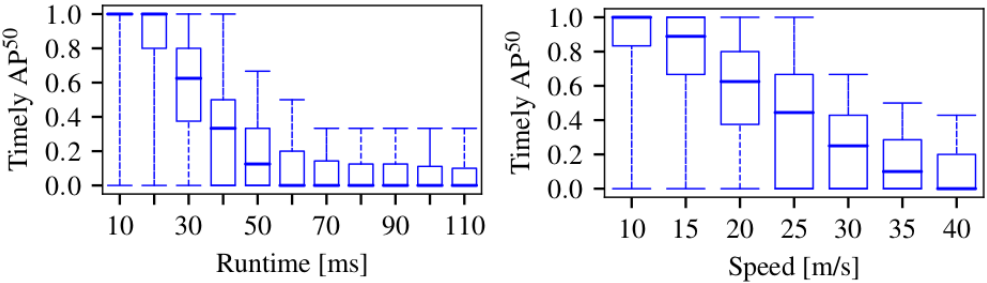


Scale: 480, Latency: 56ms

Scale: 240, Latency: 47ms

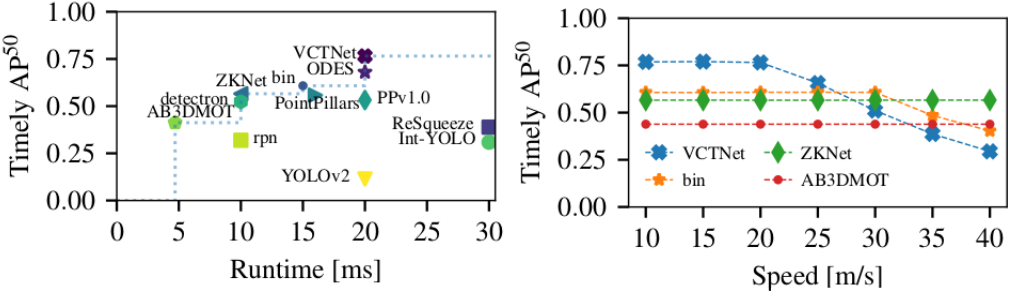
Top: FCRNN, Bottom: FCOS

Quantitative Results



Timely AP^{50} decreases with increasing run-time/speed.

Quantitative Results



Choice of the best detector changes for different run-times/speeds.

Quantitative Results

Model	Pipe.	DFP	TAL	Off AP	sAP	sAP_{50}	sAP_{75}
YOLOX-S				32.0	26.3	48.1	24.0
	✓				27.6 $\uparrow_{1.3}$	48.3	26.1
	✓	✓			28.2 (+0.6)	49.4	27.4
	✓		✓		28.1 (+0.5)	49.1	27.0
	✓	✓	✓		28.8 (+1.2)	50.3	27.6
YOLOX-M				34.5	29.2	51.9	27.7
	✓				31.2 $\uparrow_{2.0}$	51.1	31.9
	✓	✓			32.3 (+1.1)	52.9	32.5
	✓		✓		31.8 (+0.6)	53.1	31.8
	✓	✓	✓		32.9 (+1.7)	54.0	32.5
YOLOX-L				38.3	31.2	54.8	29.5
	✓				34.2 $\uparrow_{3.0}$	54.6	34.9
	✓	✓			35.5 (+1.3)	56.4	35.3
	✓		✓		35.1 (+0.9)	55.5	35.6
	✓	✓	✓		36.1 (+1.9)	57.6	35.6

Improvements from Dual-Flow Perception and Trend-Aware Loss

Quantitative Results

Li et al.	Detector	sAP	sAP_L	sAP_M	sAP_S	sAP_{50}	sAP_{75}	Runtime (ms)
Accurate *(AP)*	HTC @s1.0	*38.0*	*64.3*	*40.4*	*17.0*	*60.5*	*38.5*	*700.5*
Accurate		6.2	9.3	3.6	0.9	11.1	5.9	700.5
Fast* (Online)	RetinaNet R50 @s0.2	6.0	18.1	0.5	0.0	10.3	6.3	**31.2**
Optimized* (Online)		12.0	24.3	7.9	1.0	25.1	10.1	56.7
+S+A+F	Mask R-CNN R50 @s0.5	16.7	39.9	14.9	1.2	31.2	16.0	
+Infinite GPUs		20.3	38.5	**19.9**	4.0	**39.1**	18.9	
Ghosh et al.								
Dynamic-Online Policy		**21.3**	**47.1**	18.7	**4.4**	37.3	**21.1**	

Improvements from Reinforcement Learning

Limitations

- Metrics are new and have not been widely adopted.
- Old benchmark datasets needs to be updated for real-time applications with these metrics.
- Metrics could be biased since it relies on a scene to be constantly moving.

Summary

In real-time settings, latency-accuracy trade-off is much more important as the agent has to not only **accurately perceive** (detect/track) but also **react** (predict/plan) **as soon as possible**.

Accuracy and latency must be evaluated at the same time.

Summary

Several proposed metrics for the streaming perception task:

- **Streaming + Timely accuracy** evaluates the performance by comparing the latest available prediction with the ground-truth at current time. This forces the model to consideration of the time.
- **Dual Flow Perception + Trend Aware Loss** introduces temporal information to both training and inference pipeline.
- **Learned informative metrics** chooses the Pareto-optimal model by considering the model parameters; input resolution, and scene aggregates. These metrics are learnt with an reinforcement learning agent.

Questions?

Appendix

Hardware Benchmark Difference:
(total images/sec)

- **DFP+TAL** 2 x GTX 2080ti : 330.38
- **YOLOX** Tesla V100 : 239.86

TensorFlow: 1.12
Model: resnet152
Dataset: imagenet (synthetic)
Mode: BenchmarkMode.TRAIN
SingleSess: False
Batch size: 80 global
40.0 per device
Num batches: 100
Num epochs: 0.01
Devices: ['/gpu:0', '/gpu:1']
Data format: NCHW
Optimizer: sgd
Variables: replicated
AllReduce: None

`https://hackmd.io/@chweng/r1Ec8EXWV?type=view#2X-GeForce-RTX-2080-Ti-FP16`

Appendix: Effects of RL

Table 4: Resilience to various fixed policies

Approach	AP
Ours (s=s_1, np=np_1, R=R_2, π_{fixed} = (640, 300))	**21.3**
Ours (s=s_1, np=np_1, R=R_2, π_{fixed} = (480, 300))	**21.3**
Ours (s=s_1, np=np_1, R=R_2, π_{fixed} = (360, 1000))	**21.1**

Table 5: Resilience to choice of action spaces

Approach	AP
Ours (s=s_2, np=np_1, R=R_1, strategy=ϵ-greedy)	18.2
Ours (s=s_2, np=np_1, R=R_1, strategy=ucb)	20.0
Ours (s=s_2, np=np_1, R=R_2, strategy=ϵ-greedy)	**20.4**

Table 6: Performance with Tracking

Approach	AP
Static Policy (s=900, ts=600, k=5)	17.8
Static Policy (s=600, ts=600, k=5)	19.0
Ours (s=s_1,ts, k, np=$np1$, R=R_2)	**19.4**

Table 7: Performance with Model switching

Approach	AP
Streamer (m=$fcos$, s=600)	16.7
Streamer (m=$yolov3$, s=600)	20.2
Streamer (m=$frcnn$, s=600)	20.4
Ours (m=m, s=s_1, np=np_1, R=R_2)	**20.7**

dimensions: (1) Scale (s): s1 = 720, 640, 560, 480, 360 and s2 = 750, 675, 600, 525, 450; (2) Number of proposals (np): np = 100, 300, 500, 1000; (3) Tracker scale (ts): ts = 720, 640, 560, 480, 360; (4) Tracker stride (k): k = 3, 5, 10, 15, 30 and (5) Model choice (m): m = yolov3, fcos, frcnn.

`https://ui.adsabs.harvard.edu/link_gateway/2021arXiv210605665G/arxiv:2106.05665`