A Survey on Latency-Aware/Streaming Perception

Guray Ozgur Worakorn Ruangratanawicha

Abstract

A successful model or algorithm is said to be successful when compared with the state-of-the-art counterparts. However, the most of the time, the comparison is like comparing apples to oranges. There are several important design concepts such as accuracy, speed and memory to be taken into consideration. Accuracy and speed are usually intertwined with each other. To be able to achieve a higher accuracy, the model or algorithm has to take its time to be sure, which results in an inversely proportional relation between speed and accuracy. Moreover, a low speed algorithm naturally results in a high latency. This algorithmic trade-off between latency and accuracy has been known and both accuracy and latency have been well studied in order to obtain metrics that helps to improve them individually in the literature [13]. One of the problems is still present, and awaits to be solved. Methods are compared in traditional offline evaluations with one of the metrics. Either a high accuracy or a low latency is achieved, which leads to either accurate-but-slow or speedy-but-inaccurate methods. As a result, they fail to have good performance in real settings where both accuracy and latency play an important role. To achieve a high performance in real settings, there are studies focusing around the following ideas: (1) finding new metrics for evaluating both accuracy and latency of the pipeline/model at the same time, (2) learning the trade-offs at run-time and adapting, or designing latency-aware systems, (3) building platforms to evaluate the accuracy/latency of the methods for an end-to-end behavior.

1. Introduction

Autonomous agents can and most likely will shape various sectors ranging from logistics, transportation to medicine. There are numerous expected advantages of autonomous agents including convenience, effectiveness, and improved safety. For instance, autonomous vehicles might actually decrease the number of crashes caused by human mistakes. Furthermore, they could assist the traffic flow by reducing the congestion. There has been significant advances in AV field, in particular in all sub-tasks of the AV pipeline, such as perception, motion prediction, motion planning, and control [10]. However, development of AV pipeline depends on individual sub-task and each sub-task is evaluated with task-specific metric. These metrics do not take into account of the run-time of the model and are generally evaluated in offline settings. Even KITTI [6] and Cityscapes [2], which are datasets prepared in order to evaluate the performance of the models in driving situations, employ average precision as a metric. Hence, the latency is disregarded from the evaluation. Especially, 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. Hence, the best model (the most successful one) resides in a sweet-spot on the latency-accuracy curve [13].

In AV research, this problem is addressed as streaming perception, in which the autonomous agent is trying to perceive its environment and take action in timely manner. Another challenge is the real-time nature of the task, specifically, the current state of the environment would have changed by the time the agent finished processing, consequently, the agent would constantly be reacting to the past. This means that reliability is measured by both accuracy and end-to-end latency for applications where safety is the most crucial element. Yet, there was no metrics to address this issue.

In this manner, proposals rely the idea of evaluating models by pairing the inputs with the current state of the world. Li et al. [13] propose a metric called *streaming accuracy* to accommodate the needs of real-time settings, namely high accuracy and low latency. Their contributions include highlighting the problem again, finding the optimal for a task-specific problem, observing that tracking and forecasting are interlaced. Gog et al. [10] propose an open-source platform for exploring latency-accuracy trade-offs. In this platform, they introduce a new family of metrics, *timely accuracies*. The timely accuracies evaluate results of modules in the pipeline with respect to the present world. Thus, the trade-off between the accuracy and the run-time can be easily studied for the safety of the agent. Ghosh et al. [7] propose a method based on deep reinforcement learning where they introduce various metrics, namely *switchability*, *scale*, *aggregate* metrics, in order to choose from latency-accuracy trade-offs by jointly optimizing a reward function taking into consideration of both accuracy and latency. Moreover, the agent is dynamic, thus it can adapt itself in the online setting.

2. Background

2.1. 2D Object Detection

Existing domain-specific image object detectors are often classified into two types: two-stage detectors like Fast R-CNN [9] and one-stage detectors like YOLO [18], SSD [16], RetinaNet [14], and FCOS [22]. Backbones of earlier 2D object detectors are usually ResNet [11], VGG-16 [21] and DarkNet [19]. Two-stage detectors achieve exceptional localization and object identification accuracy, while one-stage detectors provide fast inference [12]. To evaluate the models, following metrics are used:

Average Precision (AP): The most common metric for the evaluation of models is Average Precision (AP) in object detection challenges. There are several variations of AP to be explored.

The fundamentals are:

· True Positive (TP): A correct detection of an object

· False Positive (FP): A wrong detection of an object

· False Negative (FN): An undetected object

Moreover, it is unclear to talk about a correct detection with bounding boxes. For this reason, it is common to use IOU with a threshold to decide correctness, which measures the ratio of overlapped area and union area between predicted box B_p and ground-truth box B_{gt}

$$\text{IOU} = \frac{\text{area}\left(B_p \cap B_{gt}\right)}{\text{area}\left(B_p \cup B_{gt}\right)} \tag{1}$$

Since TN is not present as mentioned, True Positive Rate (TPR), False Positive Rate (FPR) and Receiver Operating Characteristic (ROC) can not be used to evaluate. Instead, Precision (P) and Recall (R) can be used, as shown in Equation 2, 3 respectively. Precision measures the ability of identifying relevant objects, whereas recall measures the ability of identifying all ground-truth boxes.

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detections}} \tag{2}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truths}} \tag{3}$$

For varying confidence values associated with the bounding boxes created by a detector, the precision recall curve can be seen as a trade-off between precision and recall. As a result, an object detector can be deemed good if its precision remains high as the recall increases, which indicates that a large Area Under the Curve (AUC) is desirable. It is hard to obtain AUC from precision-recall curve in practice. The properties of precision-recall curve is understood by averaging over maximum precision values by using a set of recall values. This metric is called Average Precision (AP). 11-point interpolation (AP_{11}) or all-point interpolation (AP_{all}) are two of the approximations that are used. Since datasets used in object detection are hosting different classes,

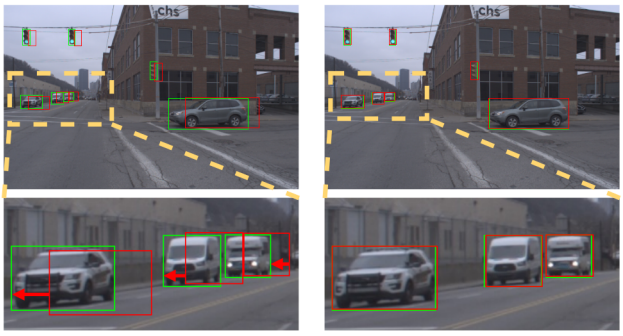


Figure 1. Visualisation of the problem. Ground truth is represented by the green boxes, whereas predictions are represented by the red boxes. The red arrows indicate the shifts should be done in the prediction boxes to help the latency problem.

the mean AP (mAP) is utilized. It is simply the average AP over all classes, as indicated in Equation 4 where AP_i is the respective AP for i^{th} class and N is the number of classes [17].

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{4}$$

Frames Per Second (FPS): The most common metric for comparing models, especially fast detectors such as Fast R-CNN [8], Faster R-CNN [20], YOLO [23] is FPS. The need of fast detectors became obvious with the PASCAL VOC [4] for real-time object detection. To rank the speed of the detectors, FPS metric is utilized in this challenge.

Latency: Latency is a measure of delay. It measures how much time it takes to process for the algorithm. It is usually measured in miliseconds.

2.2. Problem with the metrics

Fast detectors are in pursuit of the optimal latency-accuracy trade-off for real-time applications. However, to evaluate the results, two different metrics are used: one for accuracy and one for speed. That's why, it is hard to determine which latency-accuracy is the best suited detector for the task at hand. For instance, for Real-Time Object Detection on COCO [15], as of May 2022, YOLOR-D6 [24] is the leading for mAP, YOLOv4-CSP CD53s 640 [23] is leading for FPS, YOLOv4-CSP-P6 [23] is leading for inference time, ms.

Real-time applications require a prediction to the current state of the world, which is illustrated in Figure 1. From Figure 1, it can be seen that prediction done for an earlier input lags behind of the current ground-truth of the world as the the world dynamically changes. Even, as the world is dynamic, and the speed of the autonomous vehicle is different at different times, it is most likely that different world scenario's need different type of detectors. Hence, neither of the metrics used separately is not helpful for real-time scenarios. There are several proposals to address this

issue. We will talk about some of them, namely *streaming accuracy*, and *timely accuracies*. Then, we will present some applications that uses these newly introduced metrics to evaluate their model, as well as a reinforcement learning method that learns *informative metrics* in Section 3.

2.3. Streaming Accuracy

Li et al. [13] present a metric called *streaming accuracy* (sAP) to measure the joint effects of both accuracy and latency by forcing the perception pipeline to compare the prediction with the ground-truth of the current state, which directly results in the latency having influences on the accuracy scores. To able to that data needs to be timestamped. Thus, data will not be paired as an input and ground-truth but also with the timestamps, i.e. as $\{(x_i, y_i, t_i)\}_{i=1}^{T}$. The algorithm should have an access to the past observation from the current time, i.e. only $\{(x_i, t_i) \mid t_i \leq t\}$ is accessible for the time t.

The algorithm is allowed to make predictions at any time. By collecting the timestamps that the algorithm produced a prediction, for timestamps s_i, N predictions as in $\{(\hat{y}_j, s_j)\}_{j=1}^{N}$ are obtained.

Now, the ground-truth and prediction pairs can be formed. Here, by putting a real-time constraint to the predictions, the latest prediction before the current time is paired with the ground-truth as $\left\{\left(y_i, \hat{y}_{\varphi(t_i)}\right)\right\}_{i=1}^{T}$ where $\varphi(t) = \arg\max_j s_j < t$ is the real-time constraint.

The loss evaluated by the new formed pairs can be called streaming loss and it is shown in Equation 5.

$$L_{\text{streaming}} = L\left(\left\{\left(y_i, \hat{y}_{\varphi(t_i)}\right)\right\}_{i=1}^{T}\right) \tag{5}$$

Here, the inference time of the following modules is not included to the metric, however, the authors argue that the extensions of this idea is well applicable, which are taken into consideration in Section 2.4 by Gog et al. [10]. The applicability of this metric extends to any single-frame task such as object detection, instance segmentation as well as object tracking.

Furthermore, it is argued that this idea naturally combines tracking and forecasting, which are necessary to detect an object accurately in real-time scenarios considering the latency. As a result, to show the effects of increasing streaming accuracy, they add fast trackers and forecasting to the pipeline.

2.4. Timely Accuracies

In order to quantify the influence of run-time on module accuracy, Gog et al. [10] propose a new family of measures called *timely accuracies*. The timely accuracy is a metric that measures how accurate a module's findings are in the present situation of the world, not to the past situation represented in the input. The fundamental concept is to compare the output created for input at time t_1 to the ground truth at time t_2, where $t_2 = t_1 + l_t$ and l_t is the run-time of the module. Hence, in online settings, timely accuracies show what old accuracy metrics fails to show, which is how the accuracy decreases with long run-times.

They have also argued that the new metrics, timely accuracies, does not only show the effect of the run-time of the algorithms but also capture the effects of change in a dynamic environment. The effect of change in a dynamic environment is especially important for autonomous vehicles as they are in a constantly changing environment. This means that detectors or trackers behave differently under different circumstances depending on the speed of the vehicle, change in the traffic conditions etc. The newly introduced metric timely accuracy is able to show the latency-accuracy trade-off with increasing run-time of the detector and increasing speed of the vehicle for a pedestrian detector, which is a very safety-critical application.

2.5. Benchmark Datasets

Since streaming perception is an emerging topic, there has not been many datasets specific for the task, other than Argoverse-HD [13], however, the task overlap with object detection tasks, thus datasets for autonomous vehicles such as KITTI [6], BDD100K [26] and Cityscapes [2] can indicate the performance of streaming perception pipeline when adopted properly. Moreover, methods to improve streaming perception will probably be integrated to popular AV simulators such as CARLA [3].

2.6. Improvement to fast detectors from YOLOX

If something will be said about fast detectors, YOLO should be mentioned briefly no matter what. YOLO series are the responses for emerging real-time applications, the aim is both fast and accurate detectors. Ge et al. [5] propose some experienced improvements to YOLO series, and come with YOLOX whose baseline is YOLOv3. YOLOX was also the backbone for the before-mentioned work of Yang et al. [25].

YOLOX offers a superior trade-off between speed and accuracy than other competitors thanks to certain current enhanced detection approaches such as decoupled head, anchor-free, and sophisticated label assignment strategy. By improving the design of YOLOv3, which is one of the most extensively used detectors in industry, they enhanced the best AP on COCO by 3.0 percent at the time they released their model.

Furthermore, they won the first place of Streaming Perception Challenge on WAD 2021, which is a challenge that is evaluated on Agroverse HD dataset with the streaming accuracy metric. The best trade-off for latency-accuracy is found by streaming loss on 30 FPS data stream with the inference time $\leq$ 33ms. This shows that the fast detectors

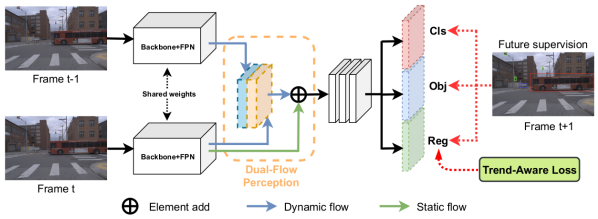


Figure 2. The training pipeline for streaming perception using Dual-Flow Perception module (DFP) and Trend-Aware Loss (TAL)

are in a good direction to real-time applications, if they are adopted well.

3. Methods

3.1. Dual-Flow Perception and Trend-Aware Loss

Yang et al. [25] argues that detectors are imposed to have a future forecasting ability instead of finding a sweet-spot along the accuracy-latency curve. At training time, a Feature Pyramid Network (FPN), a feature extractor, extracts the features from the current and previous frames. Then, collected feature maps from Dual-Flow Perception module (DFP) are sent to the classification, objectness, and regression heads. There is also a future supervision by using the ground truth of the next frame. For efficient training, an additional Trend-Aware Loss (TAL) is applied to the regression head, shown in Figure 2.

Dual-Flow Perception: Dual-Flow Perception is running two detectors on two adjacent frames, aggregating the previous feature map to the current one, and only extract features on the current frame, so that the run time is almost the same the base detector YOLOX [5].

Trend-Aware Loss: The authors argue that in streaming perception, the moving speed of each object inside a single frame is highly diverse. A Trend-Aware Loss (TAL) adjusts the weight of each object based on its movement. Fast-moving objects are examined closely as their future states are more difficult to foresee. Matching IOU is introduced as in Equation 6. This value correlates with the moving speed. The smaller value means object is moving fast, and the other way around.

$$mIoU_i = \max_j \left(\left\{ \text{IoU} \left(box_i^{t+1}, box_j^t \right) \right\} \right) \tag{6}$$

If a new object arrives, mIOU is considerably smaller since there is no corresponding box. A threshold has been established to solve this problem. Hence, the trend factor w_i is obtained for each object and its Equation is given in Equation 7.

$$\omega_i = \begin{cases} 1/mIoU_i & mIoU_i \geq \tau \\ 1/\nu & mIoU_i < \tau \end{cases} \tag{7}$$

Further, w_i is normalized to maintain the sum of total loss.

$$\hat{\omega}_i = \omega_i \cdot \frac{\sum_{i=1}^{N} \mathcal{L}_i^{reg}}{\sum_{i=1}^{N} \omega_i \mathcal{L}_i^{reg}} \tag{8}$$

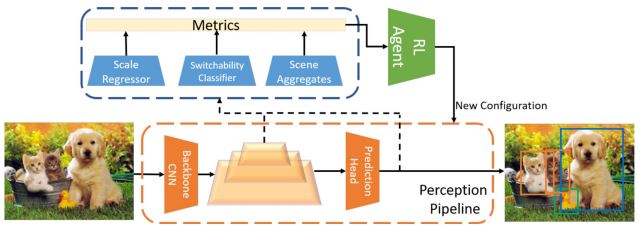


Figure 3. Overview of the approach predicting informative metrics by using a Reinforcement Learning Agent

where $\mathcal{L}_i^{reg}$ is the regression loss of object i. By weighting the regression loss with the trend factor a trend-aware total loss is obtained, as shown in Equation 9.

$$\mathcal{L}_{\text{total}} = \sum_{i \in \text{ positive}} \hat{\omega}_i \mathcal{L}_i^{reg} + \mathcal{L}_{cls} + \mathcal{L}_{obj} \tag{9}$$

3.2. Learning Informative Metrics with Reinforcement Learning

While the method discussed in Section 3.1 focuses on a novel architecture and loss function, we now discuss an orthogonal idea that tries to instead learn some informative metrics with reinforcement learning. Ghosh et al. [7] first argues that the computation cost of picking up models from latency-accuracy curve is exploding combinatorially with increasing the number of choices. Picking up does not depend on real-time data, and as mentioned it does not consider latency and accuracy together. Hence, they propose to learn the metrics for guiding trade-off decisions on run-time.

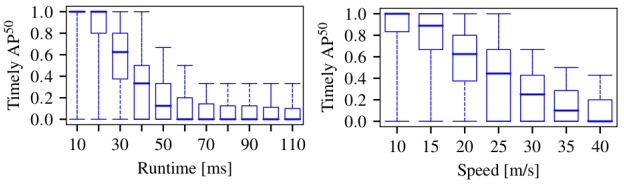
They propose to use lightweight classifiers/regressors to learn informative metrics by using feature outputs of the last convolutional layer of CNN as shown in Figure 3. The metrics are related with the latency-accuracy trade-off, such as altering the input resolution or switching between models.

- *Switchability:* The deep neural networks complexity varies depending on how deep or wide they are, leading to variable latency-accuracy trade-offs. The switchability metric attempts to represent the differences between the many models. There are three categories for switchability, namely low, medium, and high. A low switchability means switching does not make sense between models, whereas a high switchability means the result is highly effected when the model is switched.

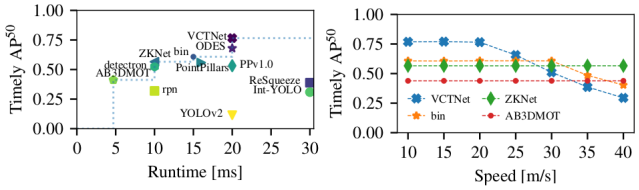
- *Adaptive scale:* Adascale is a scale metric that measures the effect of input resolution, and aims to find the optimal input resolution so that both accuracy and latency are improved.

- *Scene aggregates:* Since latency-accuracy depends on the scene, they include some frame-level aggregates such as:

  · Confidence aggregates
  · Category aggregates



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



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

Figure 4. The latency-accuracy trade-off is illustrated with a timely accuracy for a detector.

· Object size aggregates
· Crop extents

They design an online reward function that meets both accuracy and latency criteria. The reward function compares the model output to a ground truth stream of the real world state in an online method as in Li et al [13].

To design a reward function, two actions of the agent need to be compared. Let a_1 be the action taken by the agent at time t_{a_1} and a_2 at time t_{a_2}.

$$R\left(t_{a_1}, t_{a_2}\right) = L\left(\left\{y_k, \hat{y}_{\varphi(t_k)}\right\}_{k=t_{a_1}}^{t_{a_2}}\right) \tag{10}$$

However, what the reward function in Equation 10 is missing that for sequences that are inherently challenging, the agent may choose the optimal model but get a smaller reward. Hence, a fixed policy reward $R_{\pi_{fixed}}\left(t_{a_1}, t_{a_2}\right)$ must be subtracted to keep the balance as in Equation 11.

$$\hat{R}\left(t_{a_1}, t_{a_2}\right) = \lambda\left(R\left(t_{a_1}, t_{a_2}\right) - R_{\pi_{fixed}}\left(t_{a_1}, t_{a_2}\right)\right) \tag{11}$$

4. Experimental Results

4.1. Results from Li et al. [13]

Since there was no annotated dataset available for autonomous driving for streaming perception, Argoverse 1.1 with high-frame-rate sensor data (30 FPS) is annotated and ArgoverseHD is created. For the ablation of the dataset, it is compared with MS COCO [15] and found that is comparable with MS COCO, and even harder for detection of small objects. Then, Li et al. examine the performance of the state-of-the-art detectors in streaming perception task. The results are summarized in Table 1, where it clearly shows streaming perception is a challenging task, and the gap in performance with the offline evaluation continues to be significant (20.3 versus 38.0 in AP).

4.2. Results from Gog et al. [10]

Gog et al. not only emulated a simulation for the runtime-accuracy trade-off but also a 20ms simulation for object recognition and changed the driving speed to highlight the influence of an self-driving cars' speed on timely accuracy. When traveling at 10m/s, the median timely AP_{50} is 1, but it drops to 0 at 40m/s, which is illustrated in Figure 4a. They also used models from the KITTI [6] pedestrian detection competition to demonstrate the trade-off between model accuracy and runtime, as illustrated in Figure 4b. So, timely accuracy decreases with increasing run-time of the detector and increasing speed of the vehicle, which shows that latency is hard-coded in the metric.

4.3. Results from Ghosh et al. [7]

Ghosh et al. compare their technique to a number of state-of-the-art policies, including two static (offline) and one dynamic policy (offline), furthermore, if compared with Li et al., it can be seen from Table 1, in some cases, while detecting small or large objects, it performs better. By comparing AP_{75}'s, it can also be argued it adapts better to streaming data.

4.4. Results from Yang et al. [25]

Here to differentiate the performance of the models, briefly, we will remind that YOLOX won the first place of Streaming Perception Challenge on WAD 2021. Note that the work of Ghosh et al. builded upon reinforcement learning was in the second place. The quantitative results of YOLOX is given in Table 2 with the improvements of Yang et al. by adding Dual-Flow Perception Module and Trend-Aware Loss (TAL) to YOLOX baseline. The modified based detector trained on proposed pipeline by Yang et al. shows 3% sAP results improvement compared to the baseline implementation, YOLOX, as shown in Table 2. They also show that individual effects of Dual-Flow Perception module and Trend-Aware Loss for different size models.

5. Comparisons of Metrics

Streaming Accuracy: Streaming accuracy compares the latest prediction to the current ground-truth. Thus, the latency of is argued to be included in the evaluation. However, the metric does not take into account the inference time of the model to offset the processing time unlike timely accuracies.

Timely Accuracies: This metric evaluates the model at a time instant against the runtime-offset to the ground-truth and the evaluation is in online setting, it also addresses the lack of inference time during evaluation.

Informative Metrics: The informative metrics are aimed to choose from configurations of models, by getting the accuracy feedback from the ground-truth. The selection is a learning task as whole. The added complexity is in

Table 1. Performance of existing SOTA detectors for streaming perception. @ refers to the input scale. * refers to GPU image pre-processing. S+A+F refers to Scheduling + Association + Forecasting.

Li et al. [13]	Detector	sAP	sAP_L	sAP_M	sAP_S	sAP_{50}	sAP_{75}	Runtime (ms)
Accurate (Offline)	HTC @s1.0 [1]	*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 [14]	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 [9]	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. [7]								
Dynamic-Online Policy		**21.3**	**47.1**	18.7	**4.4**	37.3	**21.1**	

Table 2. The effect of the proposed pipeline, DFP, and TAL.

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

formulating the reward functions to the problem. However, it does not contribute to a general comparison of the models.

6. Limitations

As the new metrics introduced have not been widely adopted, many models are not directly comparable. The common metrics used for detectors are still Average Precision and FPS or latency, which might not be representative of the streaming perception. Moreover, most models are not evaluated in an online fashion, which is one of the key aspects for streaming perception. One of the challenges is to update the old benchmark datasets for real-time applications and adapt the evaluation metrics meeting the needs of both latency and accuracy. Creating new benchmarks that accommodate both simulators and real deployments is also crucial.

For the new metrics, evaluation requires some pipeline modifications, as such, it cannot be easily evaluated. Nevertheless, the metrics can summarize the accuracy and latency well. However, it still could be biased since it relies on a scene to be constantly moving. There could be some edge cases wherein objects are intermittently static and moving, and the metrics might not address this cases.

7. Conclusion

Streaming perception is one of the emerging tasks in autonomous vehicles research, requiring the agent to react to its environment both accurately and spontaneously [13]. Due to these requirements, accuracy and latency must be evaluated at the same time, as the speed and accuracy are intertwined. To this day, there has not been much research focusing on the issues, that's why metrics are utilised not jointly but separately.

Now, there has been several proposed metrics for the streaming perception task: *A streaming accuracy* [13], which evaluates the performance by comparing the latest available prediction with the ground-truth situation when the algorithm is finished, *timely accuracies* [10], which evaluates the performance by comparing the prediction with the ground-truth situation considering also the inference time, *learned informative metrics* [7], which evaluates the performance of model by considering the model parameters, input resolution, and scene aggregates by learning these metrics with an reinforcement learning agent.

Future work includes finding an evaluation metric, or a reward function to be satisfied in the online real-time setting, of the AV pipeline in an end-to-end matter. In the case of online real-time setting, the model configuration will change dynamically/on the fly with agent's own decisions. If this achieved, understanding this black-box algorithm, i.e. understanding the driving behaviour of the self-driving car, is probably going to be a future research topic.

References

[1] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. *CoRR*, abs/1901.07518, 2019. 6

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016. 1, 3

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 3

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 2

[5] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 3, 4

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 3, 5

[7] Anurag Ghosh, Akshay Uttama Nambi, Aditya Singh, Y. V. S. Harish, and Tanuja Ganu. Adaptive streaming perception using deep reinforcement learning. *CoRR*, abs/2106.05665, 2021. 1, 4, 5, 6

[8] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 2

[9] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. 2, 6

[10] Ionel Gog, Sukrit Kalra, Peter Schafhalter, Matthew A. Wright, Joseph E. Gonzalez, and Ion Stoica. Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles. *CoRR*, abs/2104.07830, 2021. 1, 3, 5, 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[12] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *CoRR*, abs/1907.09408, 2019. 2

[13] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception, 2020. 1, 3, 5, 6

[14] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 2, 6

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 2, 5

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. 2

[17] Rafael Padilla, Sergio Netto, and Eduardo da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 07 2020. 2

[18] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. 2

[19] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. 2

[20] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. 2

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 2

[22] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019. 2

[23] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. *CoRR*, abs/2011.08036, 2020. 2

[24] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. *CoRR*, abs/2105.04206, 2021. 2

[25] Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. Real-time object detection for streaming perception, 2022. 3, 4, 5

[26] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2018. 3