Real-time Streaming Perception System for Autonomous Driving

Yongxiang Gu1,2, Qianlei Wang3, Xiaolin Qin1,2,*
1Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu, Sichuan, China
2University of Chinese Academy of Sciences, Beijing, China
3Key Laboratory of Flight Techniques and Flight Safety, Civil Aviation Flight University of China, Guanghan, Sichuan, China

*Corresponding author (E-mail: qinxl2001@126.com)

Abstract—Nowadays, plenty of deep learning technologies are being applied to all aspects of autonomous driving with promising results. Among them, object detection is the key to improve the ability of an autonomous agent to perceive its environment so that it can (re)act. However, previous vision-based object detectors cannot achieve satisfactory performance under real-time driving scenarios. To remedy this, we present the real-time streaming perception system in this paper, which is also the 2nd Place solution of Streaming Perception Challenge (Workshop on Autonomous Driving at CVPR 2021) for the detection-only track. Unlike traditional object detection challenges, streaming perception task requires achieving a balance of accuracy and latency rather than focusing on the absolute performance, which is crucial for real-time autonomous driving. Based on YOLOv5, we adopted data augmentation, Bag-of-Freebies, and Transformer to improve streaming object detection performance with negligible extra inference cost. On the Argoverse-HD test set, our method achieved 33.2 streaming AP (34.6 streaming AP verified by the organizer) under the required hardware. Its performance significantly surpasses the fixed baseline of 13.6 (host team), demonstrating the potentiality of application.

Keywords—Object detection; Streaming Perception; Deep Learning; Autonomous Driving; Transformer

I. INTRODUCTION

Autonomous driving is a comprehensive task that consists of multiple perception level subtasks [1]. In the process of autonomous driving, detecting all potential dangerous targets of the complex surrounding traffic scene in a fast and accurate manner is the basis of safe driving. Meanwhile, detecting traffic signs and road environment is the premise of route planning. Therefore, object detection is a long-term research hotspot in autonomous driving [2].

Thanks to the development of deep neural networks (DNNs), modern object detection approaches have been fundamentally evolved. Currently, object detection algorithms are widely used in various applications. For traffic sign recognition classification, He et al. [3] proposed a lightweight convolutional network consisting of simple convolution and pooling layers, achieving high accuracy. Lv et al. [4] combined feature pyramid network (FPN) [5] with SSD for improving the small object detection performance of remote sensing images. Wu et al. [6] designed a unified framework composed of transfer learning, segmentation, and active classification for vehicle detection. Meanwhile, the emerging research of object detection is developing rapidly. Liu et al. [7] introduced Transformer into backbone and reduced model complexity by limiting the receptive field of multi-head attention operations to the fixed, achieving the state-of-the-art performance of 60.6 AP on COCO test-dev [8].

Nevertheless, most approaches merely focus on pushing the standard offline evaluation performance. While for practical application scenarios, accuracy and latency need to be considered comprehensively. To attract researchers to pay more attention on this issue, Streaming Perception Challenge is proposed to score latency and accuracy through streaming AP [9]. As briefly shown in Fig. 1, the real-time processing ability of detection algorithms for 30 FPS continuous video stream is tested with streaming AP. Specifically, participants are required to develop an efficient object detector on Argoverse-HD [10] dataset to achieve higher streaming AP. Since such metric is hardware-dependent, the final score is tested under the official required hardware, i.e., Tesla V100.

We highlight our principal contributions as follows:

1) Based on the YOLOv5 model, we used data augmentation, Bag-of-Freebies, and Transformer for improving the streaming accuracy with negligible extra inference cost. Furthermore, we get a real-time streaming perception system.

2) Experiments on Streaming Perception Challenge show the efficiency of our proposed method, which achieved 33.2 streaming AP (34.6 streaming AP verified by the organizer) and significantly surpassed the fixed baseline of 13.6.

II. RELATED WORK

A. Modern Object Detection

Fig. 2. Structure of modern object detectors

Fig. 1. Streaming perception evaluation
As shown in Fig. 2, a modern detector is usually composed of the following parts: a backbone which outputs the feature map of the whole picture, a neck which fuses feature maps with different scales to obtain multi-scale information, and a head which is used to predict classes and bounding boxes of objects based on the proposals [11]. In general, modern object detection framework can be mainly divided into two categories: one-stage and two-stage. The major difference between them in structure is that the one-stage generates proposals directly on the feature map, while the two-stage uses RPN to assist in generating proposals.

For two-stage detectors, Faster RCNN [12] is classical and has become the de facto standard architecture, which integrates feature extraction, proposal extraction, bounding box regression, and classification into one network, greatly improving the comprehensive performance. However, the extra sub-network of region proposals (RPN) requires great computational resources, thus limiting the processing speed. By contrast, YOLO is a family of one-stage detectors that can provide good and real-time performance. Therefore, it is nature to consider using the YOLO algorithm to cope with the streaming perception task.

B. YOLOv5

YOLOv5 [13] is the latest version of YOLO series, which surpasses its predecessors in speed and accuracy. It is proposed by Ultralytics LLC in 2020 and is still in development. By August 2021, the latest release version of YOLOv5 is v5.0, officially called YOLOv5-P6. There are 4 default configurations, i.e., YOLOv5s6 (small), YOLOv5m6 (medium), YOLOv5l6 (large), and YOLOv5x6 (extra), in which parameters increase in sequence. Their network architecture is uniform, while there are differences in network depth and width. In this paper, YOLOv5m6 is chosen to build a baseline for the consideration of model complexity.

As shown in Fig. 3, the overview of YOLOv5m6 can be divided into three parts, i.e., backbone, neck, and detection. In backbone, many effective designs of YOLOv4 and YOLOv3 have been inherited, e.g., spatial pyramid pooling (SPP) [14] module and cross-stage partial (CSP) block [15]. The structure of C3 is consistent with CSP block. By splitting the feature map in calculation, it can significantly reduce computation overhead and optimize the gradient updating process. Specifically, Batch Normalization and SiLU are incorporated into C3 for further enhancing the learning ability. Besides, parameter-less SPP module, which only consists of a series pooling layers with different scales, is adopted to extract multi-scale content information. Therefore, there is little effect on running speed. In neck, PANet [16] is adopted to replace vanilla FPN [17] for fusing features. Comparing with FPN, an extra bottom-up path augmentation and the adaptive feature pooling are proposed in PANet for boosting information flow and aggregating features. In detection, its design is consistent with YOLO algorithm, except that the anchor size is calculated through clustering in advance. There are 4 heads in the detection part, which are used for detecting objects with different scales. What’s more, YOLOv5 also absorbed other advanced technologies. Among them, Mixup and Mosic [18] are used for data augmentation, GIoU [19] loss is used for enhancing the perception of the instance scale. It is really an excellent work that contributes a lot to the community development.

However, there exist few training tricks provided in YOLOv5, which can also boost performance and own practical significance for engineering projects. To remedy this, we systematically introduce the ignored training techniques in this paper. Besides, we introduce Transformer in the last block of YOLOv5 for enhancing the ability of extracting global information.

III. METHODOLOGY

Considering the restricted hardware and detection speed requirement, we adopt YOLOv5m6 as our basic model. Bag-of-freebies reported in YOLOv4 [18] impressed us a lot, which can improve the detection performance without increasing the inference cost. Therefore, we mainly focus on techniques in the training stage. Besides, we introduce Transformer in the last block of backbone to enhance the global perception ability at a low cost.
A. Data Augmentation

| Dataset       | Training Images | Offline AP |
|---------------|-----------------|------------|
| Agroverse HD  | 39384           | 34.0       |
| Agroverse_10per| 3959            | 29.7       |

In the Argoverse-HD dataset, there exist 8 annotated object classes: person, bicycle, car, motorcycle, bus, truck, traffic light, and stop sign, which are highly related to driving scenarios. Since images are extracted from dozens of videos, it can be found that there exist plenty of approximate frames. We believe that existing training images can provide limited supervision information and models tend to be overfitting when directly trained. To test this conjecture, we extract every 10th frame and then build a small dataset, named Agroverse_10per. As shown in Tab. I, we can get close offline results with the same quick research configuration (details will be introduced in the next section) even one-tenth of training images are used.

To increase sample diversity, we incorporate COCO [8] and BDD100K [20] as external datasets. The former is a common-type dataset and the latter is a dataset of driving scenarios. Since the classes in Argoverse-HD is the subset of COCO, we directly extract the annotated information of 8 classes. However, there is no class named stop sign in BDD100K. To solve this problem, we first extract its parent-class traffic sign and then predict stop sign using previous trained detectors. For other common 7 classes, we directly extract the corresponding annotated information from BDD100k. For the sake of rigor, we manually review the annotated information of stop sign. Finally, we get a strong augmented dataset as our final training dataset, whose statistics are shown in TABLE II. Apart from directly augmenting samples with external datasets, we adopt Mosaic [18] and Mixup in the training stage for dynamic data augmentation.

B. Data Resample

![Label Statistics of datasets](image)

As shown in Fig. 4, there exists a class imbalance in the datasets, which is a common issue in objects detection tasks. To deal with this, we design two solutions for data resampling:

1. We randomly select samples of each class in proportion, and the proportion is inversely proportional to the label number of each class.
2. We use a weighted loss function to make the contribution of each class the same in training. With the same quick research configuration, we got 15.4 and 30.8 AP when using solution one and solution two, respectively. We suspect that the failure of solution one may lie in that the learning rate is set improper when the loss function is multiplied by the weight coefficient.

C. Structural Re-parameterization

![Diagram of structural re-parameterization](image)

To build a large structure for stronger learning ability as well as larger function space in training, and convert the structure to the small one in inferencing, we use structural re-parameterization [21] in this paper. The mechanism of structural re-parameterization is effective and is easy to implement. Consider that there is a basic convolution module without a nonlinear activation function. In training, an identity branch with 1×1 convolution is added to the basic convolution module with 3×3 kernel. While in inferencing, the 1×1 convolution is regarded as a special 3×3 convolution, which is padded with zero. Therefore, two branches can be merged by adding the parameters in their respective positions. What’s more, the normalization layer and the activation function can be added normally at the end of this operation. The converting process is shown in Fig. 5.

![Comparison with structural re-parameterization](image)

TABLE III.

| Structural re-parameterization | Params | GFLOPS |
|--------------------------------|--------|--------|
| ✚                             | 35.4M  | 15.3   |
|                               | 44.8M  | 65.9   |

Specifically, we add an identity branch with 1 × 1 convolution to each convolution module to conduct structural re-parameterization and set the initialization value to zero. As shown in TABLE III, under the YOLOv5m6 configuration, we can significantly reduce the number of parameters and calculation with structural re-parameterization in inferencing.

D. Transformer

Transformer was originally developed in natural language processing (NLP), and has been recently applied to computer vision tasks to achieve the state-of-the-art results [22]. Different from convolution operations, which extract features within a local fixed field, Transformer can adaptively aggregate similar features from a global view using self-attention mechanism. The form of standard self-attention operation can be formulated as follows:
where \(X \in \mathbb{R}^{H \times W \times d}\) is the reshaped input feature map; \(H, W\) are length and width, and \(d\) is the channel dimension; \(f_{\mathcal{O}}(\cdot), f_{\mathcal{K}}(\cdot), f_{\mathcal{V}}(\cdot)\) are the linear transformation functions; \(Q, K, V\) are abbreviations of query, key and value, respectively.

In this paper, we use the standard Transformer layer in the last block of backbone to obtain global information (C3TR block in Fig. 3). Since the computational complexity of Transformer is squared with the size of the feature map, our design allows the model to enjoy the benefits of Transformer at a low cost. What’s more, other layers of the backbone can reuse the pre-trained weights of the standard YOLOv5m6. Therefore, the training efficiency can be improved.

E. Lookahead Optimizer

Optimization algorithms contribute a lot to the development of deep learning. In 2019, the author of Adam proposed Lookahead optimizer [23], and proved that the performance of Lookahead is better than SGD and Adam through experiments. In Lookahead, fast weight is used to do a quick search and slot weight is used to integrate exploration and exploitation. In Lookahead, fast weight is used to do a quick search and slot weight is used to integrate exploration and exploitation. Therefore, the gradient updating process can be more stable and efficient. In this paper, we take Adam as the base optimizer and further build the Lookahead optimizer. Step number and alpha value are set as 5 and 0.5, respectively.

IV. EXPERIMENTS

A. Dataset and Evaluation

![Fig. 6. Examples of Argoverse-HD dataset](image)

| TABLE IV. | STATISTICS OF ARGOVERSE-HD DATASET |
|-----------|-----------------------------------|
| Classes   | Train Set | Validation Set | Test Set  |
|-----------|-----------|----------------|-----------|
| 8         | 39384     | 15062          | 12507     |

Based upon the autonomous driving dataset Argoverse 1.1 [24], Argoverse-HD dataset is built for streaming evaluation, in which there are a variety of road conditions (Fig. 6). Its image FPS and annotation FPS are both 30, while the image resolution is fixed at 1200×1920. There are totally 250K boxes in Argoverse-HD, therefore, models can get well trained and response detection performance credibly. For doing quick researches, we train models on Argoverse_10per (one-tenth of the train set) to test the proposed technologies. In Streaming Perception Challenge, additional datasets or annotations are allowed to train the model. Therefore, our submitted model is trained on the augmented training set described in the previous section. Since annotations of test set are kept in the organizer, our submitted model is chosen based on the performance of validation set.

For evaluation, there are two different ways, i.e., offline AP (averaged over IoU values from 0.50 to 0.95 with a step size of 0.05), and streaming AP (sAP) [9]. The former is the commonly used metric, which is hardware-independent and is evaluated on validation set, while the latter is hardware-dependent and is used for evaluation in Streaming Perception Challenge. The formal definition of sAP is shown as follows:

\[
\text{sAP}_{\text{AP}}(\phi) = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{t} \mathbb{1}\{\hat{y}_i \neq y_i, t \leq t_i\}}{\sum_{t} \mathbb{1}\{y_i \neq y_i, t \leq t_i\}}
\]

where \(x, y, t\) is the set of input image, annotation and timestamp; \(\hat{y}_i\) is the timestamp when a particular prediction \(\hat{y}_i\) is produced by the model; \(\phi(t)\) is the index of the most recent prediction at time \(t\).

In Streaming Perception Challenge, the real-time processing ability of detection algorithms is tested with sAP. The test toolkit sends frames to the object detection model in real-time and queries for the state of the world at predefined time steps (determined by FPS). Specifically, FPS of input stream is 30 and test hardware platform is Tesla V100. In this case, a detection algorithm with good sAP has the ability to work efficiently in real autonomous systems.

B. Implementation Details

In this paper, YOLOv5m6 is chosen as the baseline model. All experiments are conducted on 4 NVIDIA 2080ti GPU and the total batch size is set as 16. In the training stage, Mosaic and Mixup are used for data augmentation. Between them, Mosaic is adopted all time, while the using probability of Mixup is set as 0.24. In the testing stage, we set half precision for acceleration at Tesla V100. When doing quick researches, we only use Agroverse_10per dataset and set image resolution as 720×1280. The number of epochs is set 50. After verifying the effectiveness of all proposed technologies, we use the augmented training set and set image resolution as 1200×1920. To reduce training costs, we first train 100 epochs at 720×1280 resolution and then train 15 epochs at 1920 resolution to ensure the data consistency of training and testing distribution. Finally, Non-Maximum Suppression algorithm is used for post-processing.

C. Experimental Results

We first validate each presented technology by doing quick researches. As shown in TABLE V, the model incorporated with each technology can achieve better performance than the baseline. It should be noted that Transformer yields the highest offline AP improvement, which shows that the global information is crucial for object detection. As expected, the model incorporated with all technologies can achieve the best performance.
Since our improved method increases negligible extra inference cost, streaming AP improvement is consistent with offline AP improvement. After doing quick researches, we apply data augmentation, data resample, structural re-parameterization, Transformer, and lookahead optimizer into YOLOv5m6 model. Therefore, we get a real-time streaming perception system. It should be noted that streaming AP depends on not only the hardware but also the input size. The appropriate input size can help to achieve a balance between accuracy and latency. In the proposed method, we set the image resolution as the origin, i.e. 1200×1920. Under the official required hardware, our method achieves 36.1 streaming AP on validation set and 33.2 streaming AP (34.6 streaming AP verified by the organizer) on test set. As shown in TABLE VI, our method outperforms other competitive detectors by a large margin, demonstrating the potentiality of application for real-time autonomous driving.

To vividly demonstrate the streaming perception ability of our proposed method, we randomly selected 3 consecutive images at an interval of 5 frames in test set. All prediction results with confidence less than 0.25 are filtered. As shown in Fig. 7, the proposed method can detect the bus and bicycle in all frames, which are undetected by Mask R-CNN. What’s more, compared with Yolov5, the misjudgment problem can be improved. Benefitting from the global awareness of the transformer layer, the traffic light (in the middle frame) misjudged by Yolov5 can be recognized correctly.

V. Conclusion

In this paper, we propose the real-time streaming perception system for autonomous driving, which is also the 2nd Place solution of Streaming Perception Challenge for the detection-only track. To achieve a balance between accuracy and latency, we adopt YOLOv5 as our basic framework, data

---

| Table V. Comparison with Proposed Technologies Trained on Agroverse_10PER |
|-----------------------------------------|-----------------|-------------|-------------|
| Data Resample | Structural Re-parameterization | Transformer | Lookahead |
| ✓ | ✓ | ✓ | |
| ✓ | ✓ | ✓ | |
| ✓ | ✓ | ✓ | |

| Offline AP |
|-------------|
| 29.7 |
| 30.8 |
| 31.7 |
| 32.2 |
| 30.1 |
| |

| Table VI. Comparison with Other Methods |
|-----------------------------------------|
| Method | Backbone | Image Resolution | Offline AP | sAP | sAP\(_{50}\) | sAP\(_{55}\) | sAP\(_{S}\) | sAP\(_{M}\) | sAP\(_{L}\) |
| Mask R-CNN [25] | ResNet50 | 1200×1920 | - | 13.6 | 28.9 | 12.0 | 3.8 | 14.5 | 14.9 |
| Mask R-CNN+RL [26] | Swin-T [7] | - | - | 18.0 | 36.5 | 16.6 | 4.6 | 19.1 | 20.0 |
| YOLOv5 | YOLOv5m6 | 1200×1920 | 34.0 | 21.0 | 35.7 | 20.8 | 4.8 | 20.5 | 30.1 |
| Ours | YOLOv5m6+Transformer | 1200×1920 | 47.6 | 33.2 | 58.6 | 30.9 | 13.3 | 31.9 | 40.0 |

Note: sAP means streaming AP, sAP is evaluated on test set while offline AP is evaluated on validation set.
augmentation, Bag-of-Freebies, and Transformer are adopted to improve the streaming detection performance with negligible extra inference cost. Effective training tricks can boost performance and own practical significance for engineering projects. Therefore, we introduce them in detail and make the brief analysis. Besides, we introduce Transformer in the last block of backbone to enhance the ability of extracting global information at a low cost. By doing quick researches, we validate the effectiveness of each proposed technology. What’s more, results on Streaming Perception Challenge demonstrate the outstanding performance of our method.

ACKNOWLEDGMENT
This work is supported by Sichuan Science and Technology Program (2019ZDZX0006, 2020YFG0010), the West Light Foundation of Chinese Academy of Sciences, National Academy of Science Alliance Collaborative Program (Chengdu Branch of Chinese Academy of Sciences - Chongqing Academy of Science and Technology) and Science and Technology Service Network Initiative (KJSTS-QYZD-2021-21001).

REFERENCES
[1] Kiran, B. Ravi, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez, "Deep reinforcement learning for autonomous driving: A survey," IEEE Transactions on Intelligent Transportation Systems, 2021.
[2] Rashed, Hazem, Eslam Mohamed, Ganesh Sistu, Varun Ravi Kumar, Ciaran Eisign, Ahmad El-Sallab, and Senthil Yogamani, "Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline," Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2272-2280.
[3] He, Zhulong, Zhongjun Xiao, and Zhiguo Yan, "Traffic Sign Recognition Based on Convolutional Neural Network Model," 2020 Chinese Automation Congress (CAC), 2020, pp. 155-158.
[4] Suidong, Lv, Z. H. U. Lei, and Wang Wenwu, "Improving SSD for detecting small target in Remote Sensing Image," 2020 Chinese Automation Congress (CAC), 2020, pp. 567-571.
[5] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2017 pp. 2117-2125.
[6] Wu, Xin, Wei Li, Danfeng Hong, Jiaojiao Tian, Ran Tao, and Qian Du, "Vehicle detection of multi-source remote sensing data using active fine-tuning network," ISPRS Journal of Photogrammetry and Remote Sensing, 2020, pp. 39-53.
[7] Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," arXiv preprint arXiv:2103.14030, 2021.
[8] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft coco: Common objects in context." In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.
[9] Li, Mengtian, Yu-Xiong Wang, and Deva Ramanan, "Towards streaming perception," European Conference on Computer Vision, 2020, pp. 473-488.
[10] Chang, Ming-Fang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang et al, "Argoverse: 3d tracking and forecasting with rich maps," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8748-8757.