YOLOX-PAI

This is the improved version of YOLOX that added to the computer vision toolbox, EasyCV.

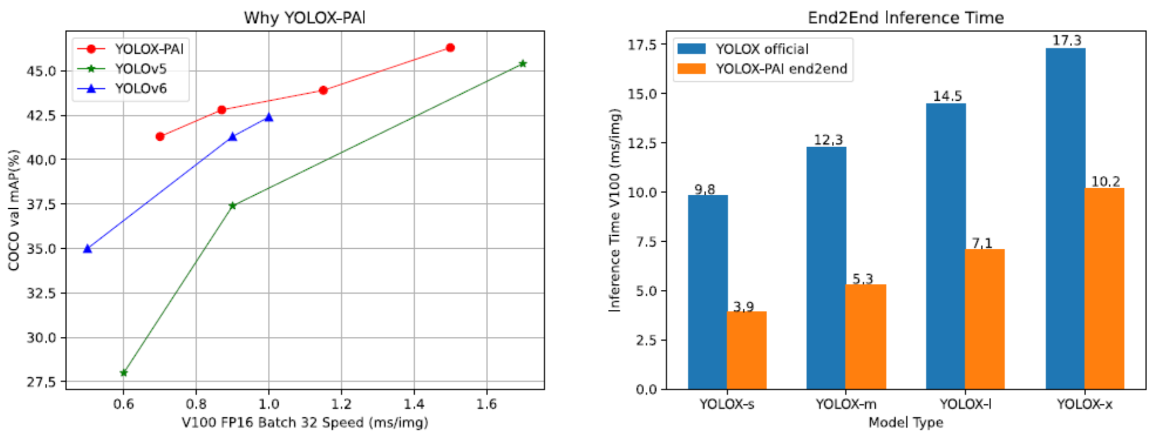


Figure 1: The comparisons between YOLOX-PAI and the existing methods.

Methods:

conducted several improvements on both the detection backbone, neck, head. also used PAI-Blade to accelerate the inference process.

Backbone:

YOLOv6 and PP-YOLOE have replaced the backbone of CSPNet to RepVGG.

In RepVGG, a 3x3 convolution block is used to replace a multi-branch structure during the inference process, which is beneficial to both save the inference time and improve the object detection results. Following YOLOv6, we also use a RepVGG-based backbone as a choice in YOLOX-PAI.

Neck:

We use two methods to improve the performance of YOLOX in the neck of YOLOX-PAI, that is 1) Adaptively Spatial Feature Fusion (ASFF) and its variance (denoted as ASFF_Sim) for feature augmentation 2) GSConv, a lightweight convolution block to reduce the compute cost.

The original ASFF method uses several vanilla convolution blocks to first unify the dimension of different feature maps. Inspired by the Focus layer in YOLOv5, we replace the convolution blocks by using the non-parameter slice operation and mean operation to obtain the unified feature maps (denoted as ASFF_Sim).

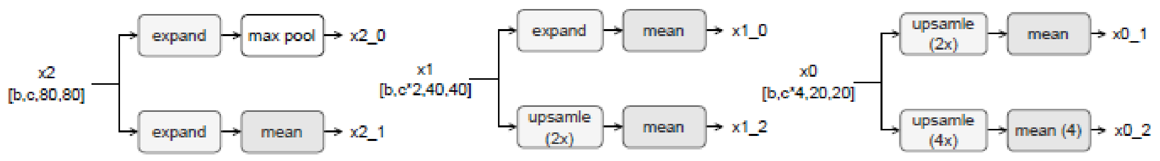


Figure 2: Operations in ASFF_Sim. The expand operation is a slice operation based on the Focus layer.

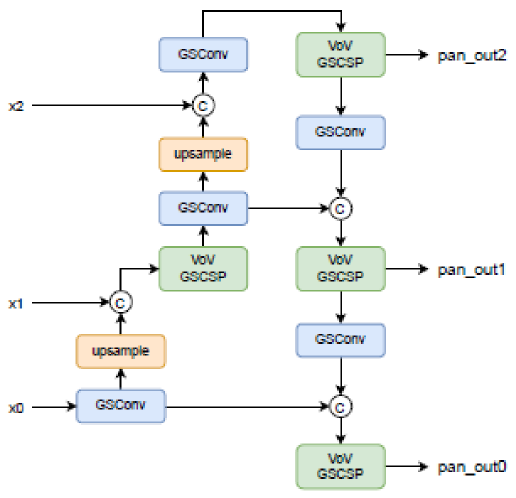


Figure 3: Architecture of the GSConv-based YOLOX neck.

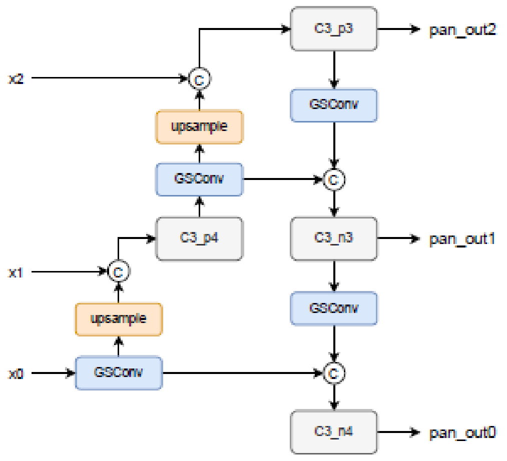


Figure 4: Architecture of the GSConv-based (part) YOLOX neck.

Head:

Enhance the YOLOX-Head with the attention mechanism as to align the task of object detection and classification (denoted as TOOD-Head). A stem layer is first used to reduce the channel, following by a group of inter convolution layers to obtain the inter feature maps. Finally, the adaptive weight is computed according to different tasks. We test the result of using the vanilla convolution or the repvgg-based convolution in the TOOD-Head, respectively.

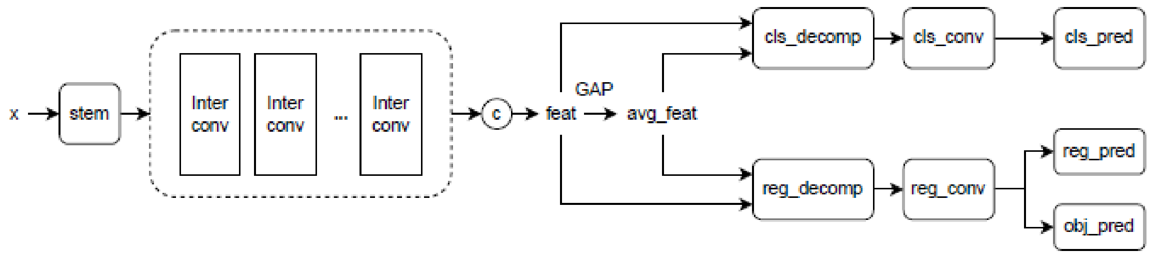


Figure 5: Architecture of the TOOD-Head.

PAI-Blade:

PAI-Blade is an easy and robust inference optimization framework for model acceleration. It is based on many optimization techniques, such as Blade Graph Optimizer, TensorRT, PAI-TAO (Tensor Accelerator and Optimizer), and so on. PAI-Blade will automatically search for the best method to optimize the input model.

the use of PAI-Blade integrated in EasyCV so that users are allowed to obtain an efficient model by simply change the export config.

EasyCV Predictor:

Along with the model inference, the preprocess function and the postprocess function are also important in an end2end object detection task, which are often ignored by the existing object detection toolbox. In EasyCV, we allow user to choose whether to export the model with the preprocess/postprocess procedure flexibly. Then, a predictor api is provided to conduct efficient end2end object detection task with only a few lines.

This model used on COCO dataset.

It needs conda virtual environment with python >= 3.6, pytorch >= 1.5, mmcv >= 1.2.0 and nvidia-dali == 0.25.0

Results:

Table 1: Comparisons between YOLOX-PAI and SOTA methods on the COCO dataset.

Model	Params (M)	Flops (G)	mAP^{val}	mAP^{val}	Speed^{V100}
			0.5:0.95 (672)	0.5:0.95 (640)	fp16 bs32 (ms)
YOLOXs [‡]	9.0	26.8	40.2	40.1	0.68
YOLOv6-tiny	15.0	36.7	41.3	-	0.9
PAI-YOLOXs	15.9	36.8	41.5	41.4	0.70
YOLOv6-s	17.2	44.2	43.1	42.4	1.0
PAI-YOLOXs-ASFF	21.3	41.0	43.3	42.8	0.87
PAI-YOLOXs-ASFF-TOOD3	23.7	49.9	44.0	43.9	1.15
YOLOXm [‡]	25.3	73.8	46.3	46.3	1.50

[‡] denotes the re-implementation YOLOX result in EasyCV, which is optimized by PAI-Blade.

Table 2: Ablation studies of ASFF and its variance.

Model	Params (M)	Flops (G)	mAP^{val}	Speed^{V100}
			0.5:0.95 (640)	fp16 bs32 (ms)
Baseline	15.9	36.8	41.4	0.70
+ASFF	21.3	41.0	42.8	0.87
+ASFF-Sim	16.3	37.1	42.6	1.13

* PAI-YOLOXs is the baseline.

Table 3: Ablation studies of GSConv block.

Model	Params (M)	Flops (G)	mAP^{val}	Speed^{V100}
			0.5:0.95 (640)	fp16 bs32 (ms)
Baseline	2.83	2.67	41.4	0.70
GSConv	1.22	1.33	41.6	0.78
GSConv_part	2.39	2.39	41.7	0.72

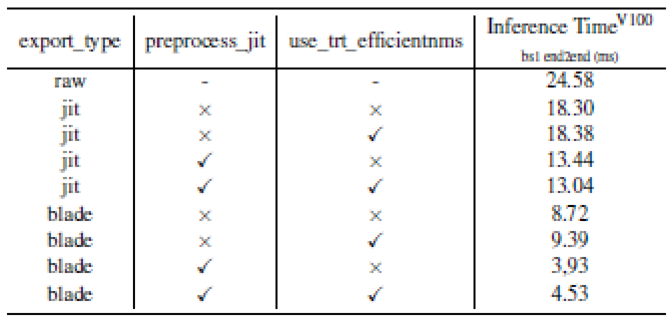
* only the parameter and flops of neck is computed.
* PAI-YOLOXs is the baseline.

Table 4: Ablation studies of TOOD-Head.

Model	Params (M)	Flops (G)	mAP^{val}	Speed^{V100}
			0.5:0.95 (640)	fp16 bs32 (ms)
Baseline	1.92	5.23	42.8	0.87
stack=2	2.22	7.71	43.5	1.09
stack=3	2.37	8.94	43.9	1.15
stack=4	2.53	10.18	44.1	1.24
stack=5	2.68	11.42	44.4	1.32
stack=6	2.83	12.66	44.7	1.40

* only the parameter and flops of head is computed.
* PAI-YOLOXs-ASFF is the baseline.

Table 5: End2end inference time of YOLOX-s with different export configs.



export_type	preprocess_jit	use_trt_efficientnms	Inference Time$^{V100}_{\text{bs1 end2end (ms)}}$
raw	-	-	24.58
jit	×	×	18.30
jit	×	✓	18.38
jit	✓	×	13.44
jit	✓	✓	13.04
blade	×	×	8.72
blade	×	✓	9.39
blade	✓	×	3,93
blade	✓	✓	4.53

Github link: https://github.com/alibaba/EasyCV

Paperswithcode link: https://paperswithcode.com/paper/yolox-pai-an-improved-yolox-version-by-pai