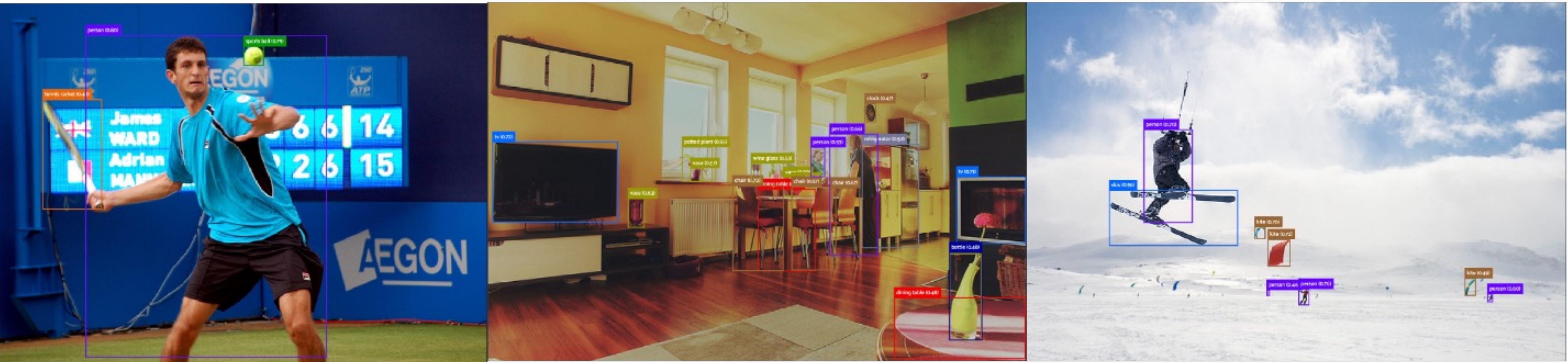
A Fast, Accurate, and Efficient Object Detection Framework

DAMO-YOLO

from Alibaba Group



TinyML Team, Data Analytics and Intelligence Lab, DAMO Academy, Alibaba Group

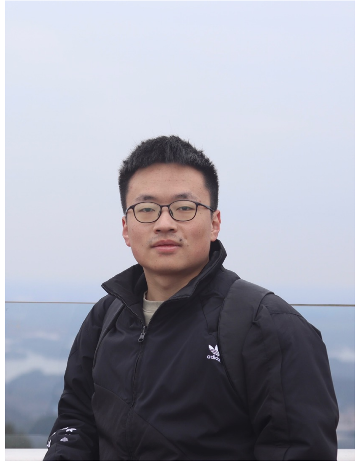
Alibaba Group 阿里巴巴集团

Contributors of DAMO-YOLO from TinyML Team

Xianzhe Xu

Yiqi Jiang

Weihua Chen



Yilun Huang

Yuan Zhang

Xiuyu Sun

Project Leader

DingTalk

WeChat

DAMO-YOLO : A Report on Real-Time Object Detection Design

Xianzhe Xu*, Yiqi Jiang*, Weihua Chen*, Yilun Huang*, Yuan Zhang*, Xiuyu Sun†
Alibaba Group

DAMO-YOLO

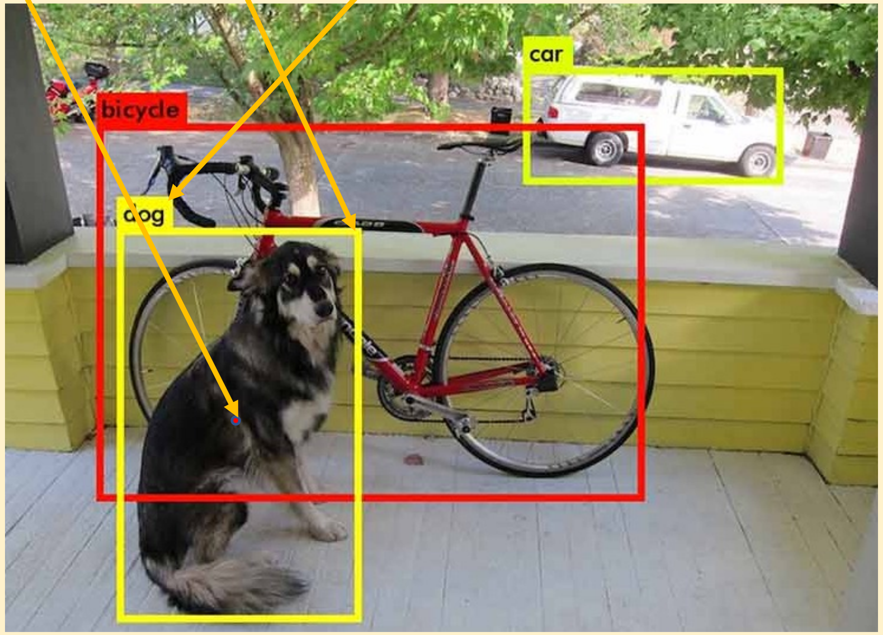
from Alibaba Group

- Introduction to Object Detection
- Recent Projects on Object Detection
- Technology Value of DAMO-YOLO
- Application Value of DAMO-YOLO
- Implementation of DAMO-YOLO

Introduction to Object Detection

Definition: locating objects of interest with their positions and sizes in images or spaces

Category



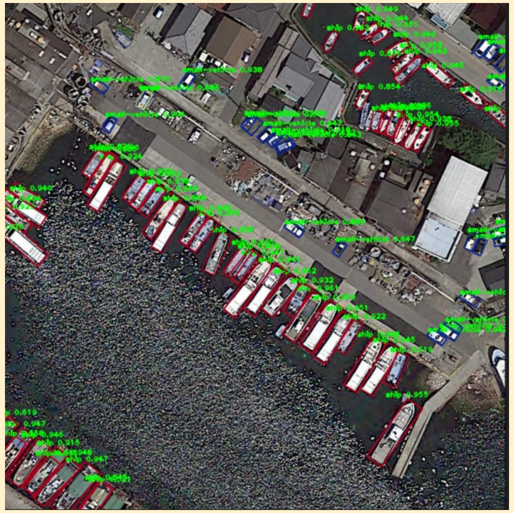
An example of object detection

Inputs: Images/Videos/Point Clouds
Outputs: Categories & Bounding Boxes

Application Value:
- Diverse application scenarios
- Fundamental task of many CV applications

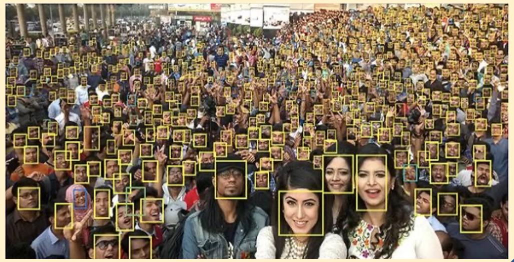
AutoPilot

Harbor Management



Intrusion Detection

Face Detection



Recent Projects on Object Detection

Current Status: Various kinds of object detection frameworks and no perfect one.

- Hundreds of models
- Popular with academia

MMDetection

PaddleDetection

- 3~5 models (10~100 GFLOPs)
- Easy to use & deploy for industry

Detectron2

YOLOX
Exceeding YOLO series in 2021

YOLOv5

YOLOv6

Flexible & Efficient

DAMO-YOLO

from Alibaba Group

Github：https://github.com/tinyvision/DAMO-YOLO
Arxiv：https://arxiv.org/abs/2211.15444
ModelScope：
https://www.modelscope.cn/models/damo/cv_tinynas_object-detection_damoyolo/summary

Disadvantages of existing frameworks:
1. Lack of flexibility in terms of model sizes: ✗ various devices with different computing power.
2. Weak multi-scale detection capability: ✗ scenarios with small objects
3. mAP-latency curve is not good enough: ✗ real-time scenarios

Tech advantages of DAMO-YOLO:
1. MAE-NAS → low-cost customizable models
2. Efficient RepGFPN+Heavy Neck → wide range of applications
3. General knowledge distillation on all sizes → improving performance without loss

Technology Value of DAMO-YOLO

DAMO-YOLO

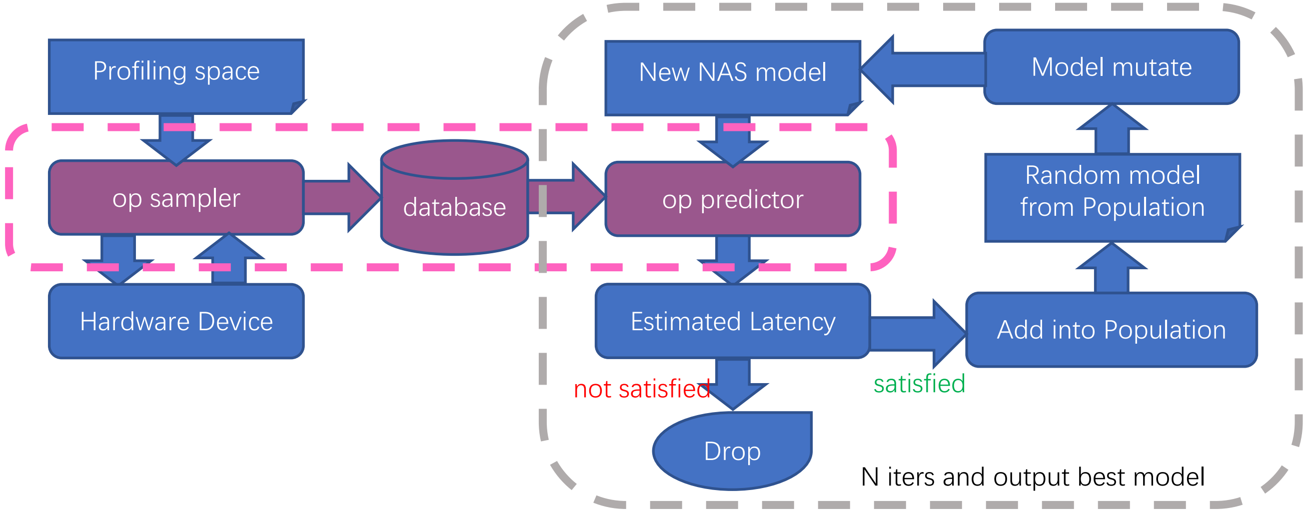
from Alibaba Group

Tech advantages of DAMO-YOLO:

1. MAE-NAS → low-cost customizable models
2. Efficient RepGFPN+Heavy Neck → wide range of applications
3. General knowledge distillation on all sizes → improving performance without loss

Ability to get low-cost customizable models:

- Search optimal model based on our MAE-NAS (ZeroShot)
  - Low-cost: no need for training
  - Improving utilization of devices: search with FLOPs/latency as budgets
- Latency database construction scheme for different devices:
  - Support T4/V100 GPU, IoT chips, etc.



Seach optimal model based on latency budget

Technology Value of DAMO-YOLO

DAMO-YOLO

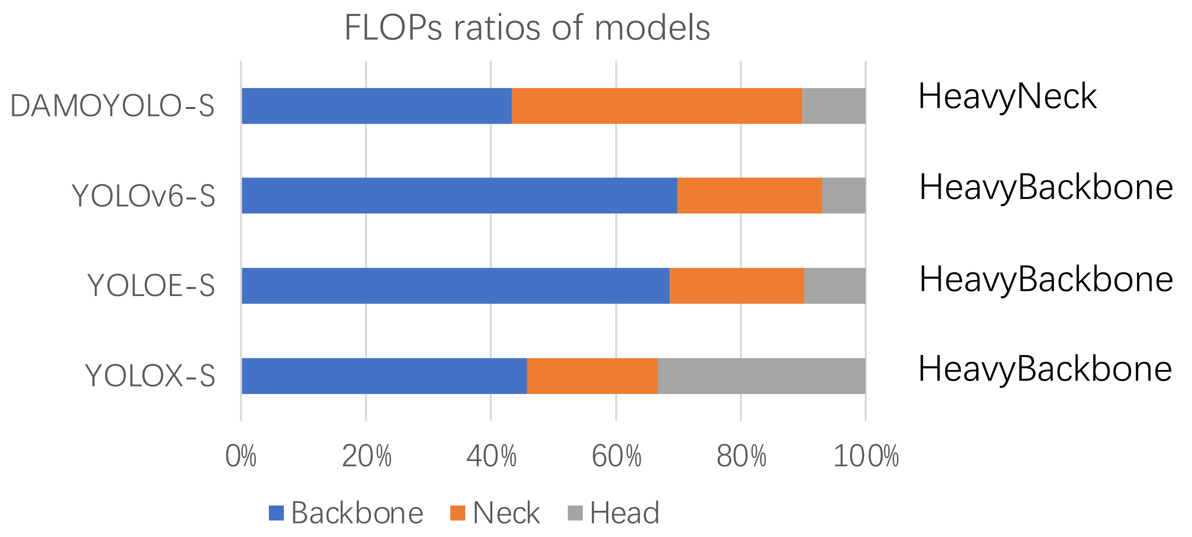
from Alibaba Group

Tech advantages of DAMO-YOLO:

1. MAE-NAS → low-cost customizable models
2. Efficient RepGFPN+Heavy Neck → wide range of applications
3. General knowledge distillation on all sizes → improving performance without loss

EfficientRepGFPN+HeavyNeck: wide range of applications

- Efficient RepGFPN: efficient multi-scale feature fusion
- HeavyNeck: redefine the FLOPs ratios of models
- Powerful multi-scale detection performance: wide range of applications



Technology Value of DAMO-YOLO

DAMO-YOLO

from Alibaba Group

Tech advantages of DAMO-YOLO:

1 MAE-NAS → low-cost customizable models

2 Efficient RepGFPN+Heavy Neck → wide range of applications

3 General knowledge distillation on all sizes → improving performance without loss

General knowledge distillation on all sizes

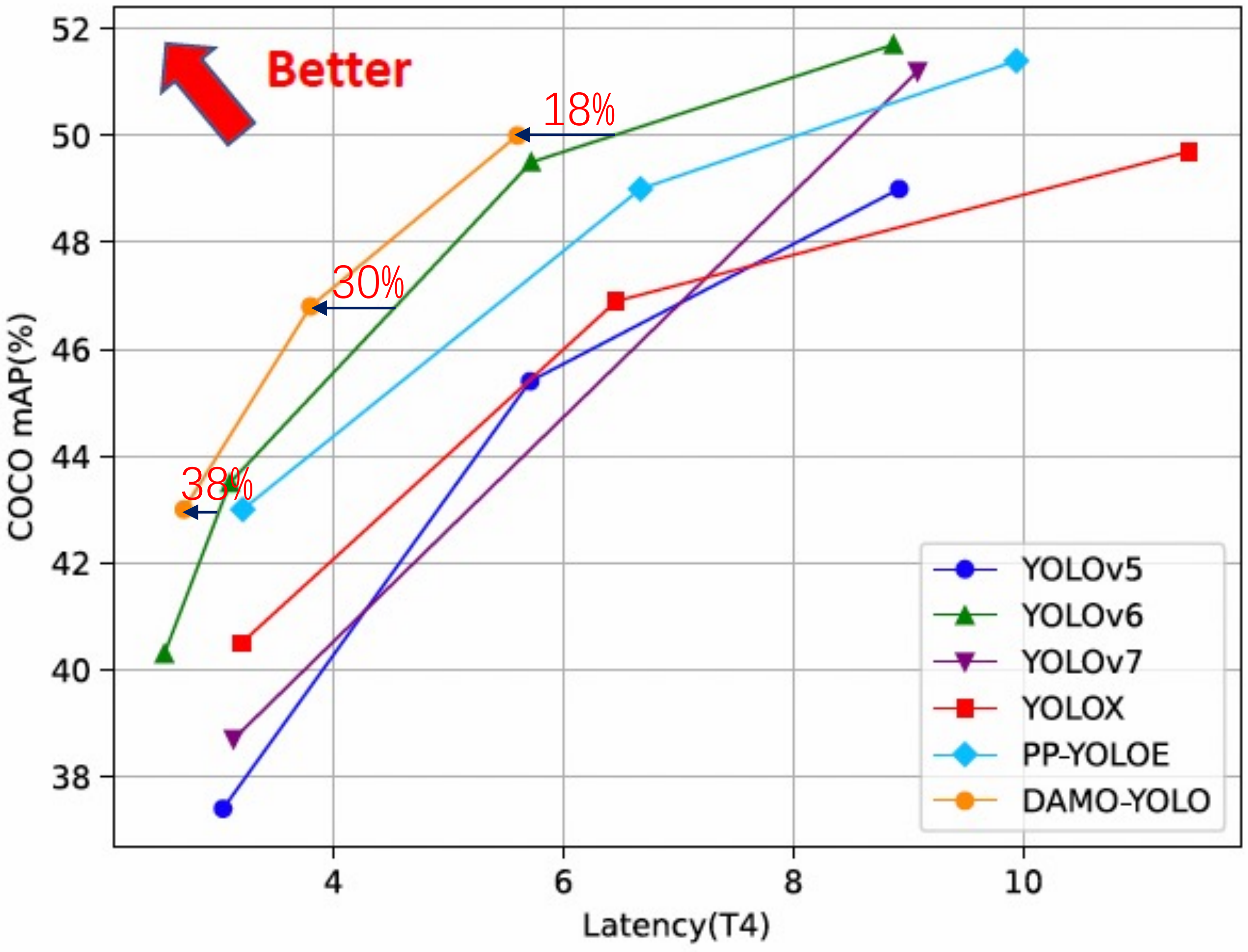
- Few studies on distillation of YOLO series in both academia and industry
- Lack of distillation scheme on small models
  - FGD: distillation on YOLOX-M
  - YOLOv6: distillation on Large and Medium sizes
- Distillation in DAMO-YOLO
  - Significant improvements on models of all sizes
  - Parameter-tuning-free: one-click script to distill
  - Feature-based distillation + unbiased BN + AlignModule: robust to heterogeneous models

Distillation performance on different model scales

Method	Size	Latency(ms)	GFLOPs	Params(M)	AP
DAMO-YOLO-T	640	2.78	18.1	8.5	41.8
DAMO-YOLO-T*	640	2.78	18.1	8.5	43.0
DAMO-YOLO-S	640	3.83	37.8	16.3	45.6
DAMO-YOLO-S*	640	3.83	37.8	16.3	46.8
DAMO-YOLO-M	640	5.62	61.8	28.2	48.7
DAMO-YOLO-M*	640	5.62	61.8	28.2	50.0

Application Value of DAMO-YOLO

Comparison in Latency-mAP trade-off



1. Speed: 20% ~ 40% faster
2. FLOPs: 15% ~ 50% less
3. Params: 6%~50% less
4. Significant improvements on all sizes & wide range of application

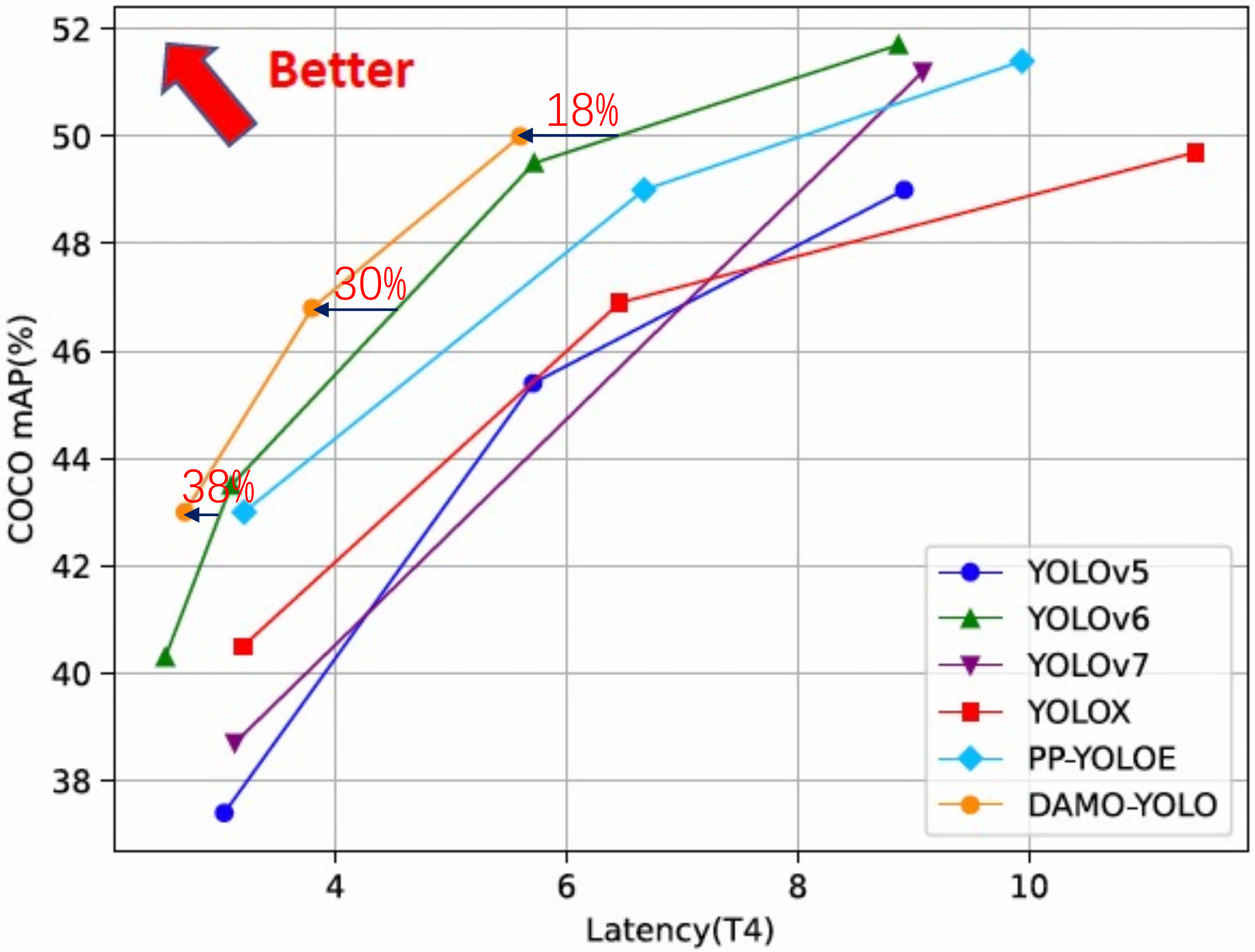
Method	Size	Latency(ms)	GFLOPs	Params(M)	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
YOLOX-T	416	1.78	6.5	5.1	32.8	-	-	-	-	-
YOLOX-S	640	3.20	26.8	9.0	40.5	-	-	-	-	-
YOLOX-M	640	6.46	73.8	25.3	46.9	-	-	-	-	-
YOLOX-L	640	11.44	155.6	54.2	49.7	-	-	-	-	-
YOLOv5-N	640	2.23	4.5	1.9	28.0	45.7	-	-	-	-
YOLOv5-S	640	3.04	16.5	7.2	37.4	56.8	-	-	-	-
YOLOv5-M	640	5.71	49.0	21.2	45.4	64.1	-	-	-	-
YOLOv5-L	640	8.92	109.1	46.5	49.0	67.3	-	-	-	-
YOLOv6-T	640	2.53	36.7	15.0	40.3	56.6	-	-	-	-
YOLOv6-S	640	3.10	44.2	17.0	43.5	60.4	-	-	-	-
YOLOv6-M*	640	5.72	82.2	34.3	49.5	66.8	-	-	-	-
YOLOv6-L*	640	9.87	144.0	58.5	52.5	70.0	-	-	-	-
YOLOv7-T-silu	640	3.13	13.7	6.2	38.7	56.7	41.7	18.8	42.4	51.9
YOLOv7	640	9.08	104.7	36.9	51.2	69.7	55.9	31.8	55.5	65.0
YOLOE-S	640	3.21	17.4	7.9	43.0	60.5	46.6	23.2	46.4	56.9
YOLOE-M	640	6.67	49.9	23.4	49.0	66.5	53.0	28.6	52.9	63.8
YOLOE-L	640	9.94	110.1	52.2	51.4	68.9	55.6	31.4	55.3	66.1
DAMO-YOLO-T	640	2.78	18.1	8.5	41.8	58.0	45.2	23.0	46.1	58.5
DAMO-YOLO-T*	640	2.78	18.1	8.5	43.0	59.4	46.6	23.3	47.4	61.0
DAMO-YOLO-S	640	3.83	37.8	16.3	45.6	61.9	49.5	25.9	50.6	62.5
DAMO-YOLO-S*	640	3.83	37.8	16.3	46.8	63.5	51.1	26.9	51.7	64.9
DAMO-YOLO-M	640	5.62	61.8	28.2	48.7	65.5	53.0	29.7	53.1	66.1
DAMO-YOLO-M*	640	5.62	61.8	28.2	50.0	66.8	54.6	30.4	54.8	67.6

Performance comparison with SOTA detectors

Application Value of DAMO-YOLO

Alibaba Group 阿里巴巴集团

Comparison in Latency-mAP trade-off



Low cost &
High efficiency

DAMO-YOLO

from Alibaba Group

1.Fast & less FLOPs: wide range of application
2.Customize models for FLOPS: improving utilization of devices
3.💥DAMO-YOLO is available on ModelScope: easy to use!

```
from modelscope.pipelines import pipeline
from modelscope.utils.constant import Tasks
object_detect = pipeline(Tasks.image_object_detection,model='damo/cv_tinynas_object-detection_damoyolo')
img_path ='https://modelscope.oss-cn-beijing.aliyuncs.com/test/images/image_detection.jpg'
result = object_detect(img_path)
```

Github：https://github.com/tinyvision/DAMO-YOLO
Arxiv：https://arxiv.org/abs/2211.15444
ModelScope：https://www.modelscope.cn/models/damo/cv_tinynas_object-detection_damoyolo/summary

DAMO-YOLO

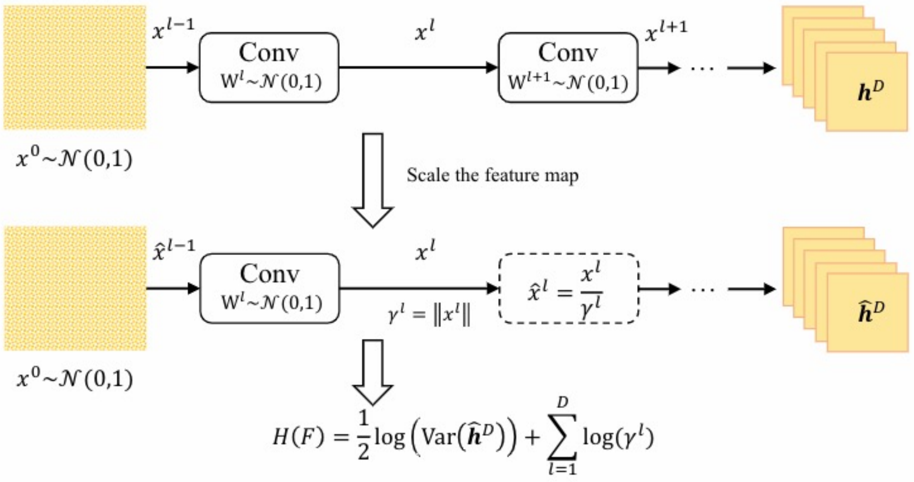
from Alibaba Group

- Implementation of DAMO-YOLO
  - **Low-cost customizable models——MAE-NAS**
  - **Efficient multi-scale feature fusion——Efficient RepGFPN**
  - **Knowledge distillation on models of all sizes**

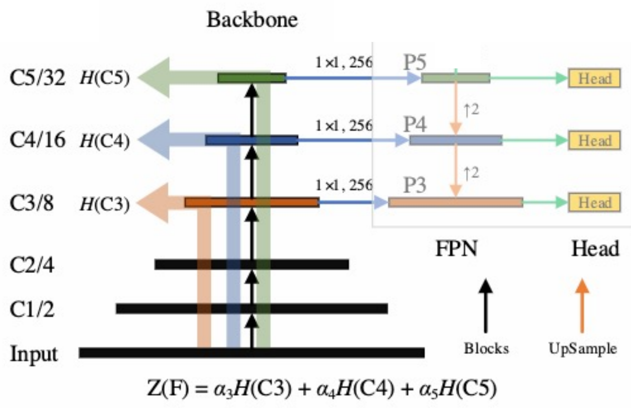
Implementation of DAMO-YOLO

Low-cost customizable models——MAE-NAS (ICML2022)

- **Idea**: formulate a deep network as an information system endowed with continuous state space, and maximize its entropy
- **Formulation**:
  - The topology of a network F can be abstracted as a graph $G = (V, E)$: vertex set V -- features, edge set E -- operators
  - $h(v), h(e)$ are values endowed with each vertex and edge. $S = \{h(v), h(e): \forall v \in V, \forall e \in E\}$ defines the continuous state space of F
  - Entropy $H(S)$ measures the total information contained in the system (network) F
    - We only focus on the feature expressivity (information contained in vertices), which is $H(S_v)$
- **Method**:
  - According to **Differential Entropy of Gaussian Distribution** and **Gaussian Entropy Upper Bound Theorem**, we can calculate feature map variance to estimate entropy $H(S_v)$ and get Gaussian entropy upper bound when they obey the Gaussian distribution
  - Therefore, all parameters are initialized by standard Gaussian distribution $N(0,1)$, and a noise image is generated with it as well
  - The (Gaussian upper bound) entropy of F: $H(F) = \frac{1}{2}\log\left(Var(\widehat{\boldsymbol{h}}^D)\right) + \sum_{l=1}^{D}\log(\gamma^l)$
  - Multi-scale entropy of F: $Z(F) \coloneqq \alpha_1 H(C1) + \alpha_2 H(C2) + \cdots + \alpha_5 H(C5)$, $\alpha = [0, 0, 1, 1, 6]$



(a) Single-scale entropy score with rescaling

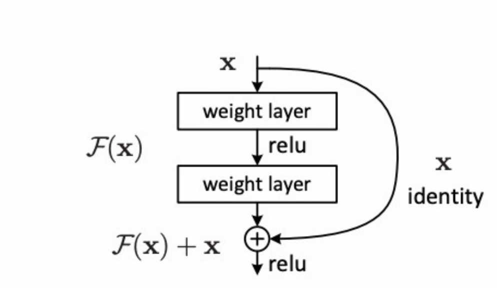


(b) Multi-scale entropy score for detection

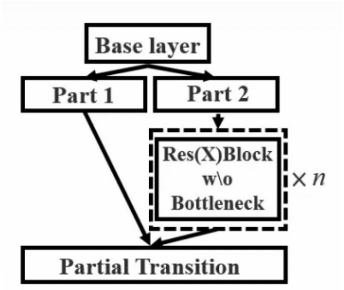
Implementation of DAMO-YOLO

Low-cost customizable models——MAE-NAS (ICML2022)

- NAS framework: Evolutionary Algorithm
  - Use multi-scale entropy of networks as the performance proxy
  - **Various search budgets:** FLOPs, Params, Latency, Layers, etc.
  - **Fine-grained mutation:** kernel size, width, depth, block type, etc.
  - **High scalability:** add customized block types easily according to the official tutorial
  - 💥**Zero-shot & GPU-free**: requires no data and no GPU, and only takes tens of minutes on CPUs
- MAE-NAS Backbone for DAMO-YOLO
  - For the goal of real-time applications, we search for T/S/M models using different latency budgets
  - Wrap the searched basic structure: smaller sizes -- ResStyle, larger sizes -- CSPStyle。



ResStyle



CSPStyle

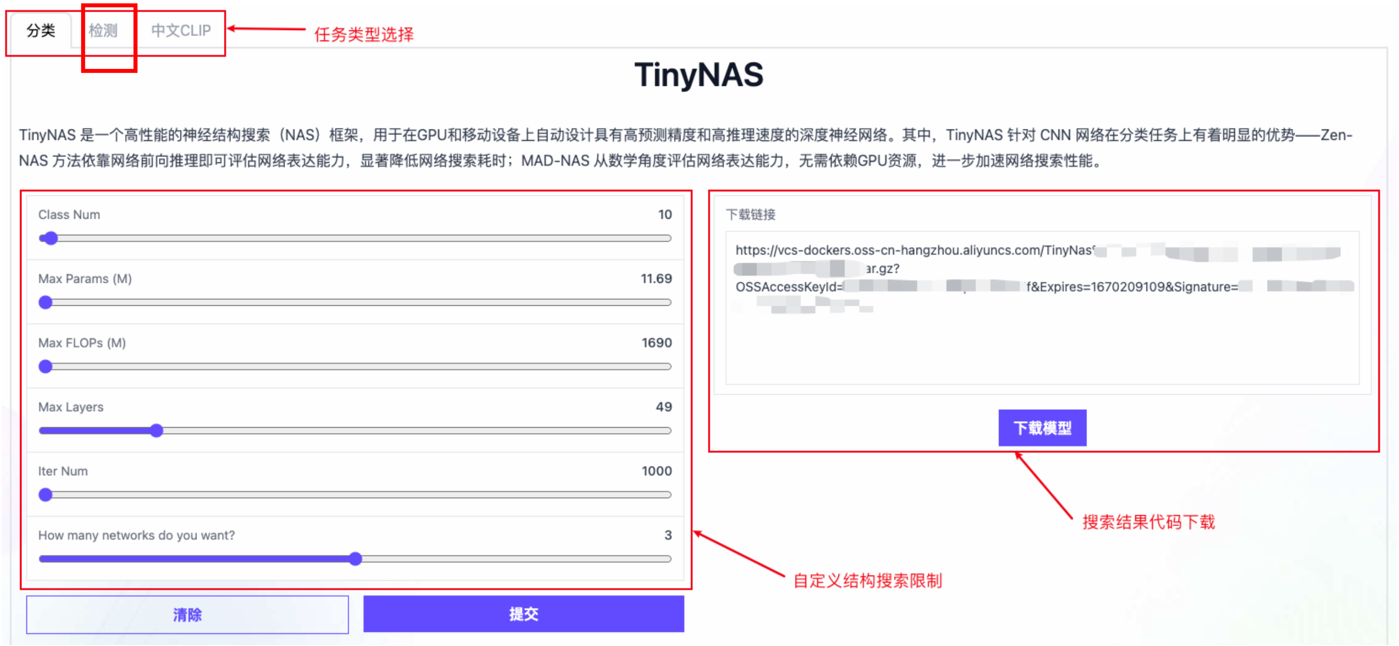
	Backbone	AP	Latency(ms)
DAMO-YOLO-S	CSP-Darknet	44.9	3.92
DAMO-YOLO-S	MAE-ResNet	45.6	3.83
DAMO-YOLO-S	MAE-CSP	45.3	3.79
DAMO-YOLO-M	MAE-ResNet	48.0	5.64
DAMO-YOLO-M	MAE-CSP	48.7	5.60

Paper: ICML2022, *MAE-DET: Revisiting Maximum Entropy Principle in Zero-Shot NAS for Efficient Object Detection*
Tutorial: NAS for DAMO-YOLO (in CN): https://github.com/alibaba/lightweight-neural-architecture-search/blob/main/scripts/damo-yolo/Tutorial_NAS_for_DAMO-YOLO_cn.md

Implementation of DAMO-YOLO

TinyNAS toolbox is available on ModelScope now!

- Based on zero/one-shot methods, you can get searched results in a few minutes
- Support various tasks and scenarios: classification, detection, Chinese CLIP
- Customizable budgets: Params, FLOPs, Layers, etc.
- Easy to load searched structures into your own networks~



Github： https://github.com/alibaba/lightweight-neural-architecture-search

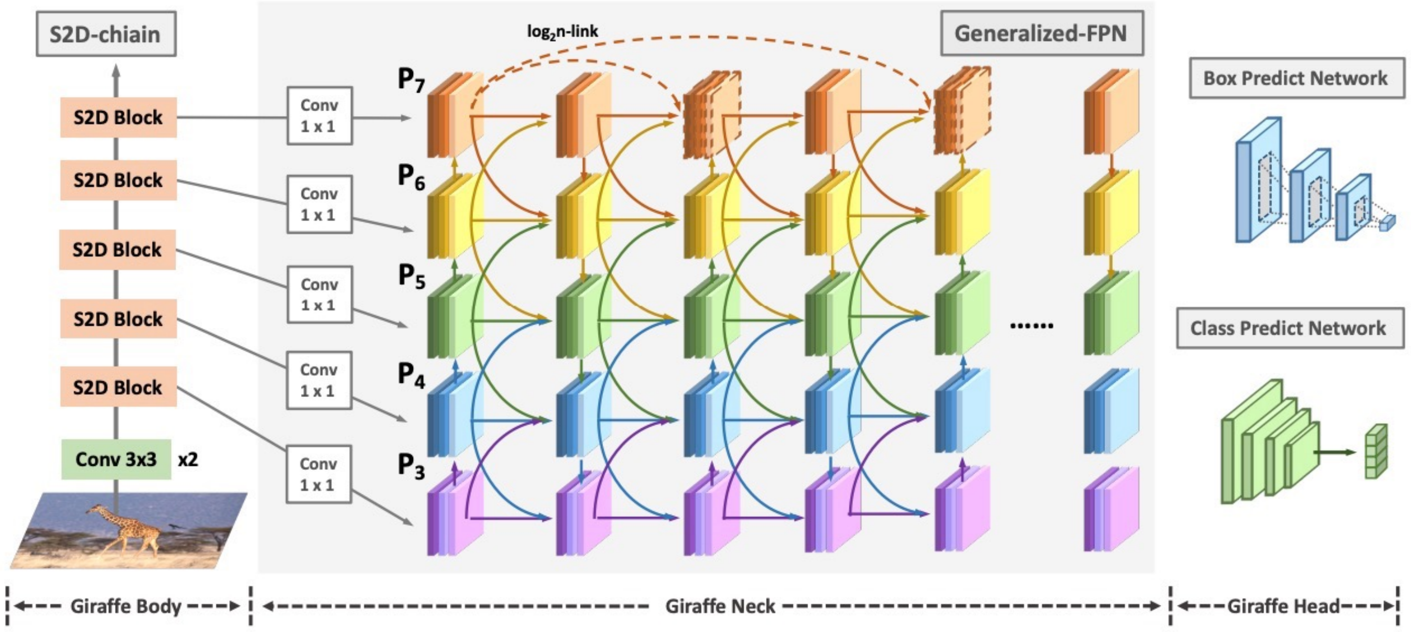
ModelScope: https://modelscope.cn/studios/damo/TinyNAS/summary

Implementation of DAMO-YOLO

Alibaba Group 阿里巴巴集团

Improving the multi-scale detection capability——GFPN (ICLR2022)

- Multi-scale detection capability depends on multi-scale feature fusion
- GFPN process high-level semantic and low-level spatial information at the same priority: beneficial to multi-scale feature fusion
- Feature reuse and more connections improve the performance, but it makes the network parallel inefficient: efficient in FLOPs but inefficient in Latency

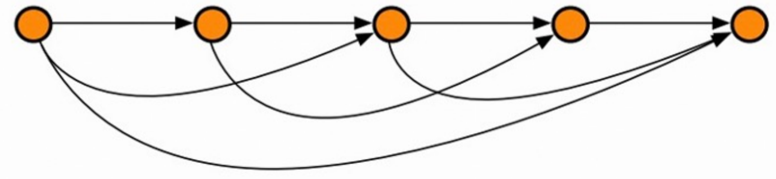


Structure of GFPN

Skip Layer

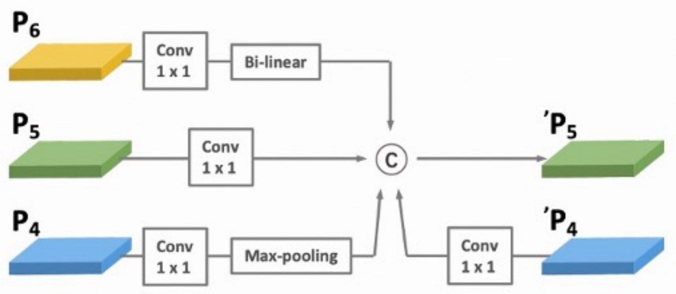
- log2n-link strengthen feature reuse and reduce redundancy

$$P_k^l = Conv(Concat(P_k^{l-2^n}, \ldots, P_k^{l-2^1}, P_k^{l-2^0})),$$



Queen Fusion

- Receive more features to improve the feature representation
  - previous P4 down, P6 up, P5, and current P4 connections



- Paper: *GiraffeDET: A Heavy-Neck Paradigm for Object Detection*, arXiv
- Code: https://github.com/damo-cv/GiraffeDet

Implementation of DAMO-YOLO

GFPN (ICLR2022) → **Efficient RepGFPN**

Existing Problem

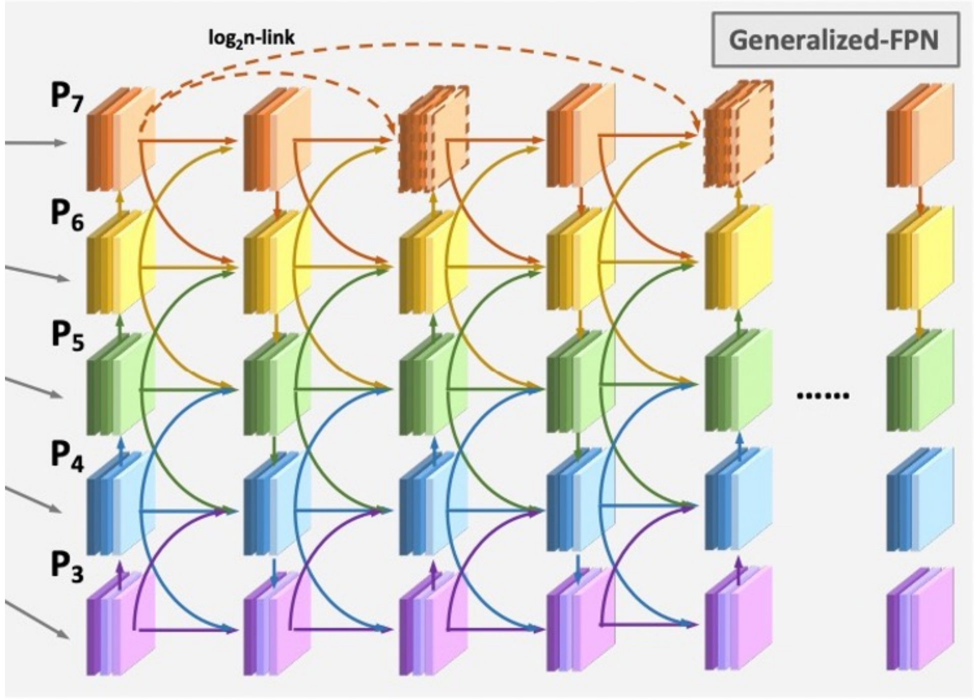
- Multi-scale features share the same num of channels
- Queen-Fusion brings inefficient connections
- Low computation efficiency in stacked nodes

Topology Optimization →

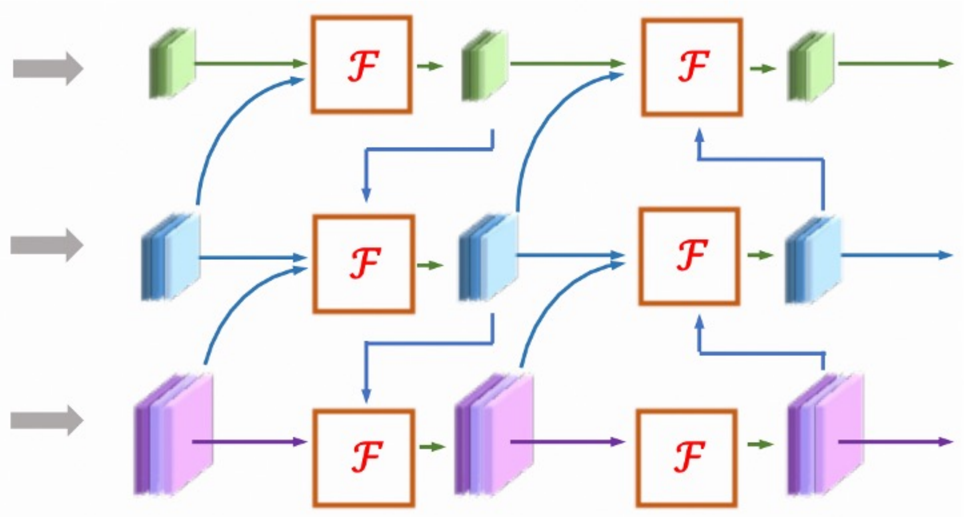
Fusion Optimization →

Optimization

- Different num of channels in different scales
- Remove inefficient up-sampling operators in Queen-Fusion
- Fix the num of nodes and optimize the fusion method



Structure of GFPN

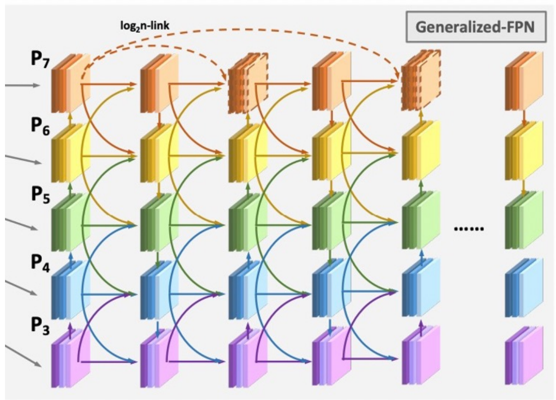


Structure of Efficient RepGFPN

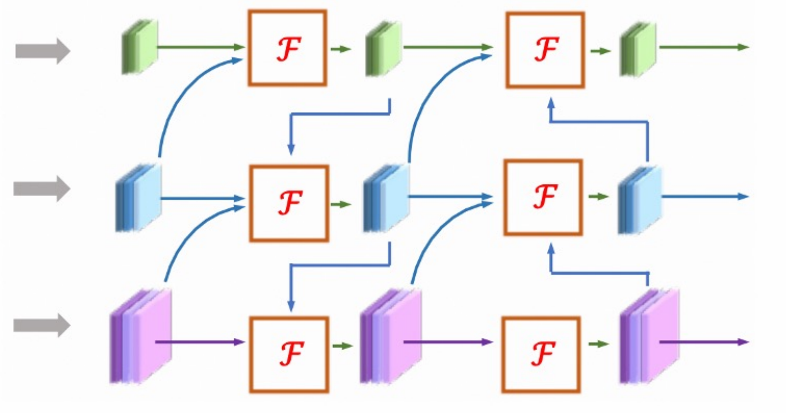
Implementation of DAMO-YOLO

GFPN (ICLR2022) → **Efficient RepGFPN**

- **Topology Optimization**
  - Multi-scale features share the same num of channels → Different num of channels in different scales
  - Queen-Fusion brings redundant connections → Remove extra up-sampling operators in it



Structure of GFPN



Structure of Efficient RepGFPN

Depth	Width	Latency	FLOPs	AP
2	(192, 192, 192)	3.53	34.9	44.2
2	(128, 256, 512)	3.72	36.1	45.1
3	(160, 160, 160)	3.91	38.2	44.9
3	**(96, 192, 384)**	**3.83**	**37.8**	**45.6**
4	(64, 128, 256)	3.85	37.2	45.3

Depth/width analysis of Efficient RepGFPN

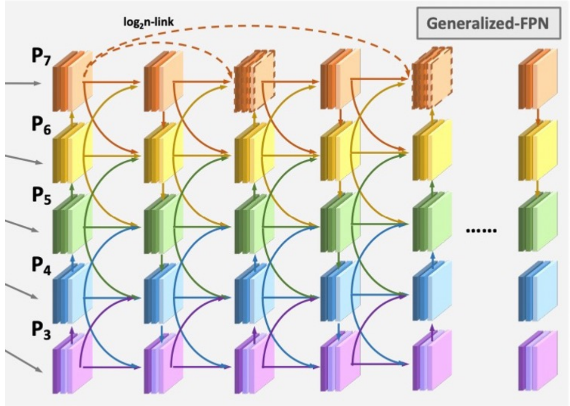
↘	↗	Latency	FLOPs	AP
		3.62	33.3	44.2
✓		4.19	37.7	44.5
	✓	**3.83**	**37.8**	**45.6**
✓	✓	4.58	42.8	45.9

Connection efficiency analysis of Queen-Fusion

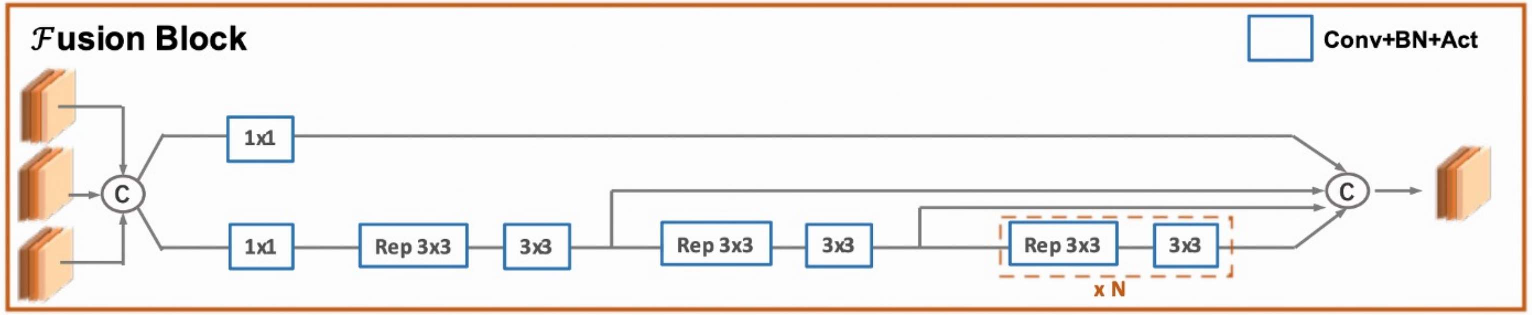
Implementation of DAMO-YOLO

GFPN (ICLR2022) → **Efficient RepGFPN**

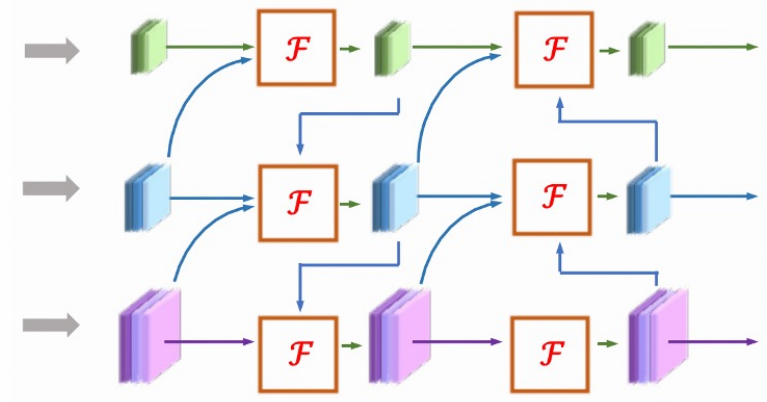
- **Fusion Optimization**
  - Low computation efficiency in stacked nodes → Fix the num of nodes, and use Fusion Block
  - CSP structure, Reparameterization, Efficient Layer Aggregation Network (ELAN)



Structure of GFPN



Structure of Fusion Block



Structure of Efficient RepGFPN

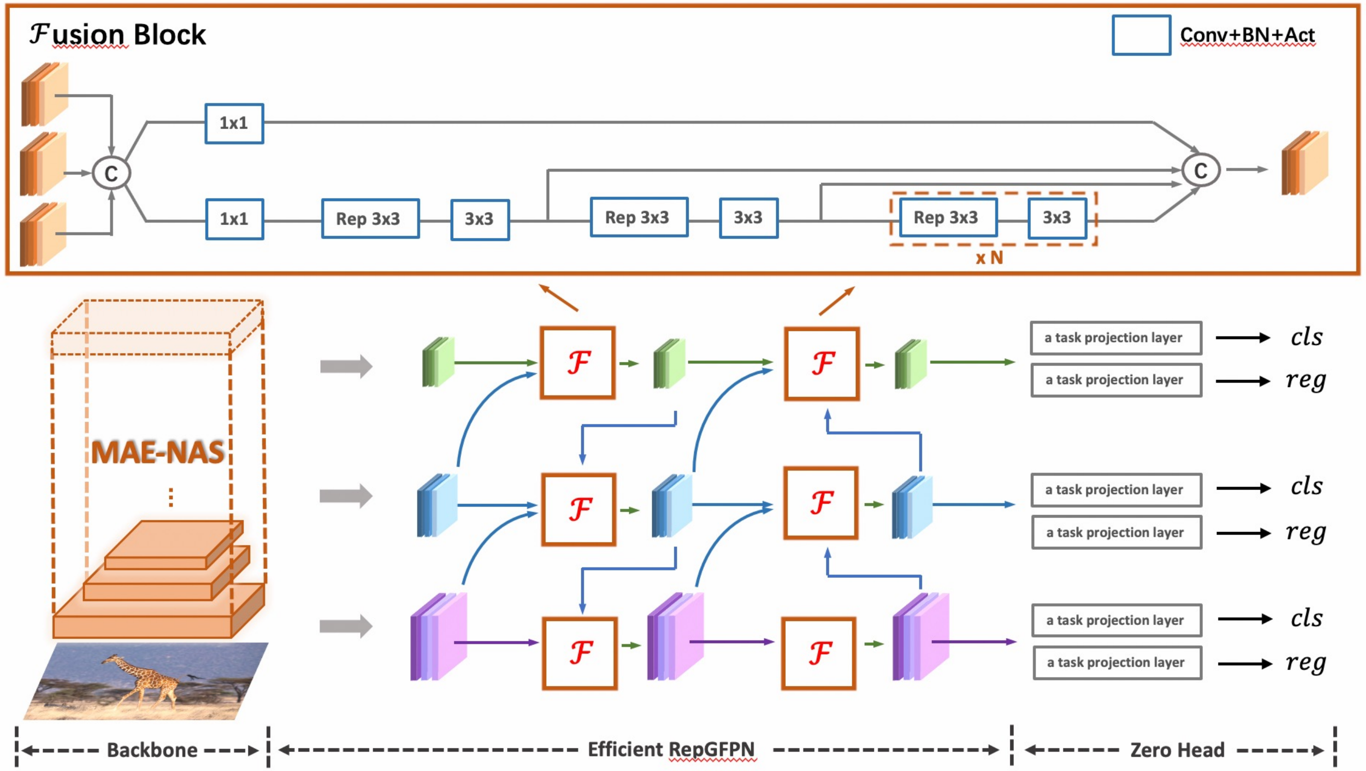
Merge−Style	Latency	FLOPs	AP
Conv	3.64	44.3	40.2
CSP	3.72	36.7	44.4
CSP + Reparam	3.72	36.7	45.0
CSP + Reparam + ELAN	**3.83**	**37.8**	**45.6**

Implementation of DAMO-YOLO

HeavyNeck & ZeroHead

- Only keep the linear projection layer for classification and regression in head
- More computations are used to stack Fusion Blocks in Efficient RepGFPN

Neck(width/depth)	Head(width/depth)	Latency(ms)	AP
(1.0/1.0)	**(1.0/0.0)**	**3.83**	**45.6**
(1.0/0.50)	(1.0/1.0)	3.79	44.9
(1.0/0.33)	(1.0/2.0)	3.85	43.7
(1.0/0.0)	(1.0/3.0)	3.87	41.2



Overall structure of DAMO-YOLO

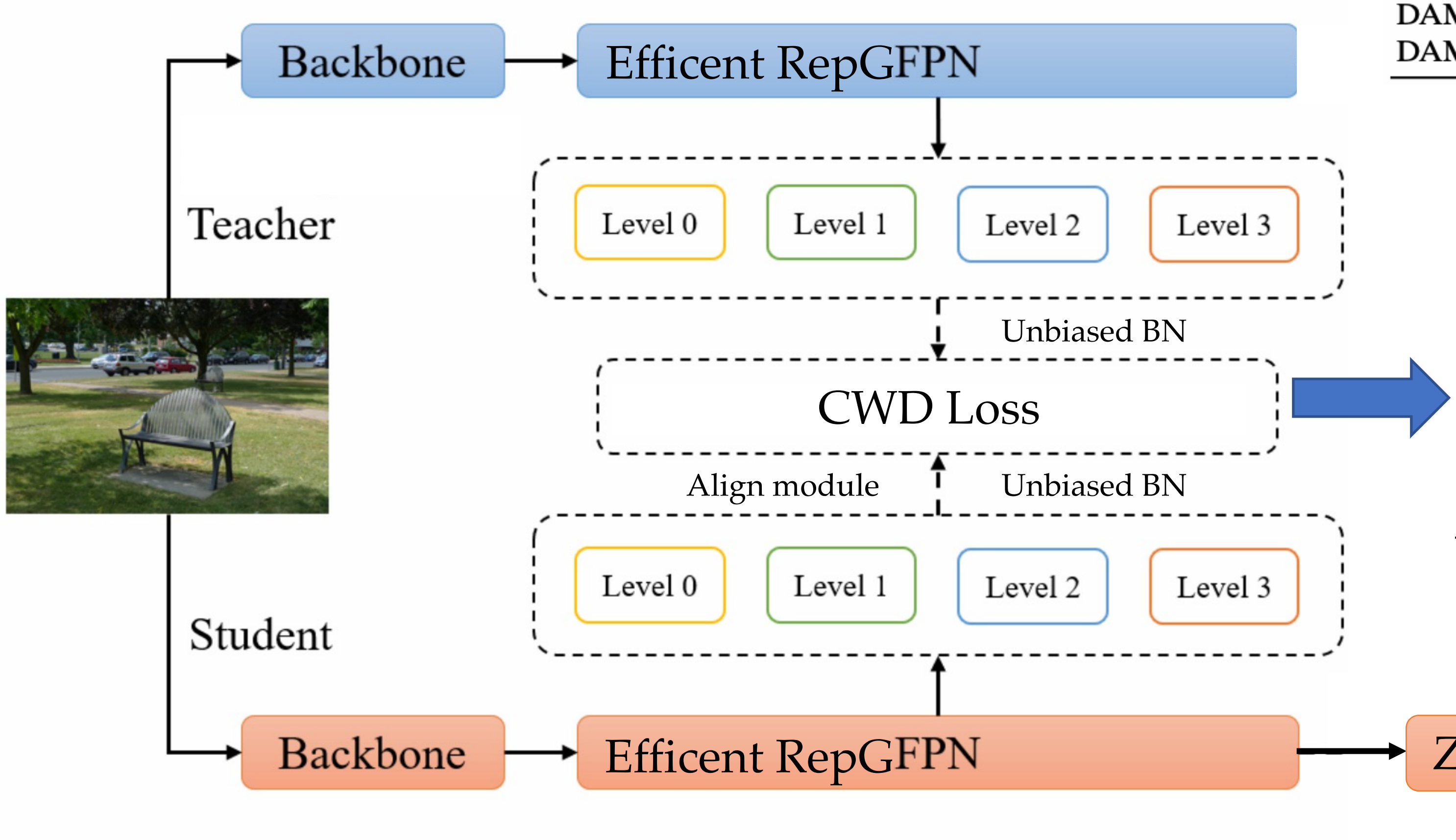
Implementation of DAMO-YOLO

Knowledge distillation on models of all sizes

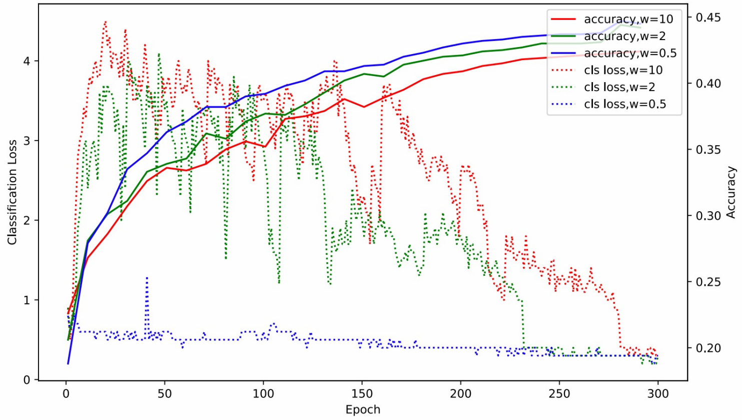
- Feature distillation+AlignModule+unbiased BN
- 1). Parameter-tuning-free, 2). Works well on all sizes, 3). Works well for heterogeneous models
- Distillation chain: L(CSP)->M(CSP)->S(Res)->T(Res)

Our distillation results on T/S/M models

Method	Size	Latency(ms)	GFLOPs	Params(M)	AP
DAMO-YOLO-T	640	2.78	18.1	8.5	41.8
DAMO-YOLO-T*	640	2.78	18.1	8.5	43.0
DAMO-YOLO-S	640	3.83	37.8	16.3	45.6
DAMO-YOLO-S*	640	3.83	37.8	16.3	46.8
DAMO-YOLO-M	640	5.62	61.8	28.2	48.7
DAMO-YOLO-M*	640	5.62	61.8	28.2	50.0



Distillation strategy of DAMO-YOLO



The cls. loss and AP curves of distillation with different weights on DAMO-YOLO-S

Implementation of DAMO-YOLO

Low cost &
High efficiency

DAMO-YOLO

from Alibaba Group

MAE-NAS

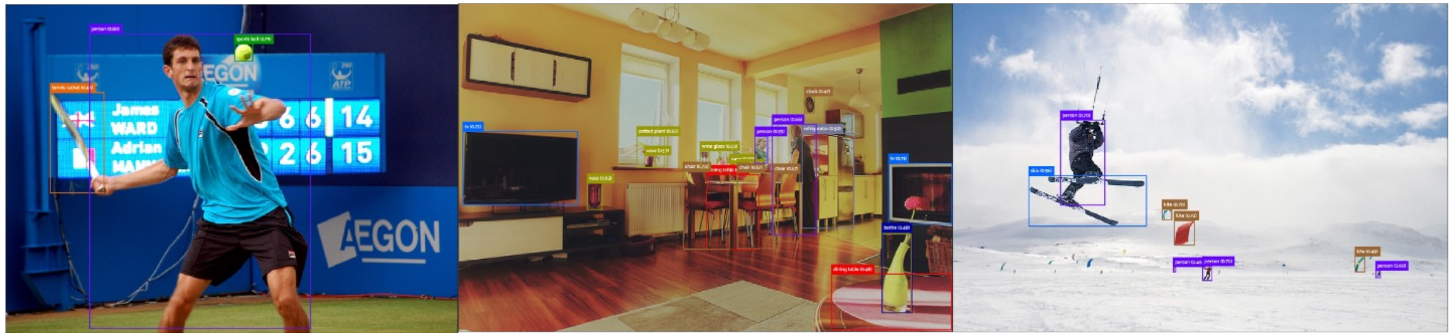
Low-cost customizable models

Efficient
RepGFPN

Efficient feature fusion &
powerful multi-scale detection capability

Distillation
on all sizes

Significant Improvement
& Easy to use



HeavyNeck
Paradigm

DAMO-YOLO is available on ModelScope now. Welcome to try it!

ModelScope：https://www.modelscope.cn/models/damo/cv_tinynas_object-detection_damoyolo/summary
Github：https://github.com/tinyvision/DAMO-YOLO
Arxiv：https://arxiv.org/abs/2211.15444

Alibaba Group 阿里巴巴集团

DAMO-YOLO

from Alibaba Group

Thanks for listening!

DAMO-YOLO is available on ModelScope. Go and try it~

https://www.modelscope.cn/models/damo/cv_tinynas_object-detection_damoyolo/summary

We are recruiting research intern, and you can send your resume to xiuyu.sxy@alibaba-inc.com

DingTalk WeChat