

ScaledYOLOv4笔记

- Paper: [Scaled-YOLOv4: Scaling Cross Stage Partial Network](#)
- Code: [ScaledYOLOv4](#)
- Blog: [Scaled YOLO v4 is the best neural network for object detection on MS COCO dataset](#)

0. Abstract

Improvements in Scaled YOLOv4 over YOLOv4

- Scaled YOLOv4 used optimal network scaling techniques to get YOLOv4-CSP -> P5 -> P6 -> P7 networks
- Improved network architecture: Backbone is optimized and Neck (PAN) uses Cross-stage-partial (CSP) connections and Mish activation
- Exponential Moving Average (EMA) is used during training — this is a special case of SWA: <https://pytorch.org/blog/pytorch-1.6-now-includes-stochastic-weight-averaging/>
- For each resolution of the network, a separate neural network is trained (in YOLOv4, only one neural network was trained for all resolutions)
- Improved objectness normalizers in [yolo] layers
- Changed activations for Width and Height, which allows faster network training
- The [net] letter_box = 1 parameter (to keep the aspect ratio of the input image) is used for high resolution networks (for all networks except yolov4-tiny.cfg)

different Losses

There are different Losses in YOLOv3, YOLOv4 and Scaled-YOLOv4:



YOLOv3	YOLOv4	Scaled-YOLOv4
$b_x = \sigma(t_x) + c_x$	$b_x = \sigma(t_x) * 1.1 - 0.05 + c_x$	$b_x = \sigma(t_x) * 2 - 0.5 + c_x$
$b_y = \sigma(t_y) + c_y$	$b_y = \sigma(t_y) * 1.1 - 0.05 + c_y$	$b_y = \sigma(t_y) * 2 - 0.5 + c_y$
$b_w = p_w e^{t_w}$	$b_w = p_w e^{t_w}$	$b_w = (\sigma(t_w) * 2)^2 * p_w$
$b_h = p_h e^{t_h}$	$b_h = p_h e^{t_h}$	$b_h = (\sigma(t_h) * 2)^2 * p_h$

Loss for YOLOv3, YOLOv4 and Scaled-YOLOv4

- for b_x and b_y — this eliminates grid sensitivity in the same way as in YOLOv4, but more aggressively
- for b_w and b_h — this limits the size of the bounded-box to $4 * \text{Anchor_size}$

network architecture

Changes to the network architecture (CSP in the Neck and Mish-activation for all layers) then eliminate flaws of Pytorch implementation, so CSP+Mish improves both AP, AP50 and FPS:

- YOLOv4-CSP — 608x608-75FPS — **47.5% AP** — **66.1% AP50**
- YOLOv4-CSP — 640x640-70FPS — **47.5% AP** — **66.2% AP50**

CSP connection is extremely efficient, simple, and can be applied to any neural network. The idea is:

- half of the output signal goes along the main path (generates more semantic information with a large receiving field)
- and the other half of the signal goes bypass (preserves more spatial information with a small perceiving field)

1. 模型结构

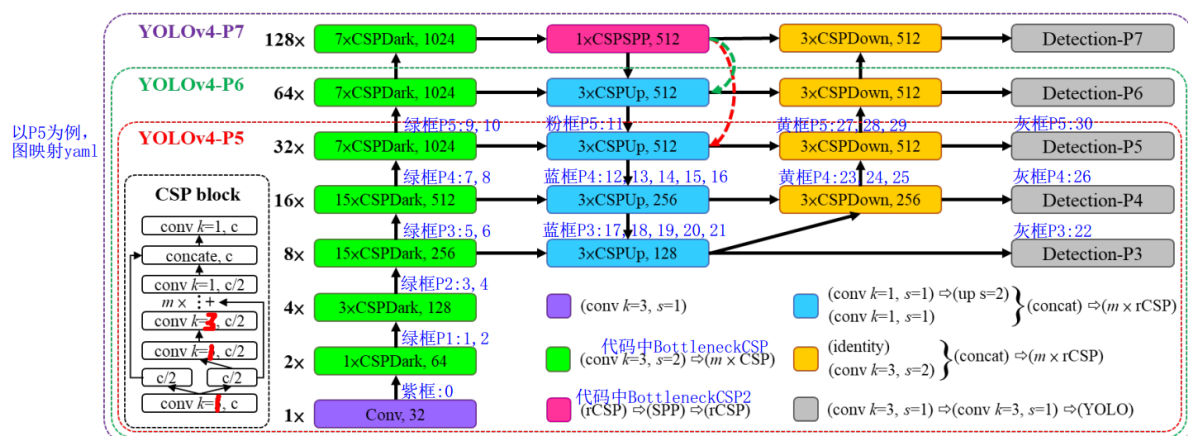


Figure 4: Architecture of YOLOv4-large, including YOLOv4-P5, YOLOv4-P6, and YOLOv4-P7. The dashed arrow means replace the corresponding CSPUp block by CSPSPP block.

- 以yolov4-p5.yaml为例

```
# parameters
nc: 80 # number of classes
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple

# anchors
anchors:
  - [13,17, 31,25, 24,51, 61,45] # P3/8 (img缩小8倍)
  - [48,102, 119,96, 97,189, 217,184] # P4/16 (img缩小16倍)
  - [171,384, 324,451, 616,618, 800,800] # P5/32 (img缩小32倍)

# csp-p5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [32, 3, 1]], # 0 (在list中的index为0) 对应图中紫框

    [-1, 1, Conv, [64, 3, 2]], # 1-P1/2 (img缩小2倍) 对应图中绿框P1中的conv
    k=3,s=2
    [-1, 1, BottleneckCSP, [64]], # 2 对应图中绿框P1中的1 x CSP

    [-1, 1, Conv, [128, 3, 2]], # 3-P2/4 (img缩小4倍) 对应图中绿框P2中的conv
    k=3,s=2
    [-1, 3, BottleneckCSP, [128]], # 4 对应图中绿框P2中的3 x CSP

    [-1, 1, Conv, [256, 3, 2]], # 5-P3/8 (img缩小8倍) 对应图中绿框P3中的conv
    k=3,s=2
    [-1, 15, BottleneckCSP, [256]], # 6 对应图中绿框P3中的15 x CSP
```

```

[-1, 1, Conv, [512, 3, 2]], # 7-P4/16 (img缩小16倍) 对应图中绿框P4中的conv
k=3,s=2
[-1, 15, BottleneckCSP, [512]], # 8 对应图中绿框P4中的15 x CSP

[-1, 1, Conv, [1024, 3, 2]], # 9-P5/32 (img缩小32倍) 对应图中绿框P5中的conv
k=3,s=2
[-1, 7, BottleneckCSP, [1024]], # 10 对应图中绿框P5中的7 x CSP
]

# yolov4-p5 head
# na = len(anchors[0]) # 每一个cell多少个anchor
head:
[[-1, 1, SPPCSP, [512]], # 11 对应图中粉框P5

[-1, 1, Conv, [256, 1, 1]], # 12 对应图中蓝框P4中的conv k=1,s=1
[-1, 1, nn.Upsample, [None, 2, 'nearest']], # 13 对应图中蓝框P4中的up s=2
[8, 1, Conv, [256, 1, 1]], # 14 route backbone P4 对应图中蓝框P4中的conv
k=1,s=1
[[-1, -2], 1, Concat, [1]], # 15 对应图中蓝框P4中的concat, 将绿框P4卷积后和上面up
的特征融合
[-1, 3, BottleneckCSP2, [256]], # 16 对应图中蓝框P4中的3 x rCSP

[-1, 1, Conv, [128, 1, 1]], # 17 对应图中蓝框P3中的conv k=1,s=1
[-1, 1, nn.Upsample, [None, 2, 'nearest']], # 18 对应图中蓝框P3中的up s=2
[6, 1, Conv, [128, 1, 1]], # 19 route backbone P3 对应图中蓝框P3中的conv
k=1,s=1
[[-1, -2], 1, Concat, [1]], # 20 对应图中蓝框P3中的concat, 将绿框P3卷积后和上面up
的特征融合
[-1, 3, BottleneckCSP2, [128]], # 21 对应图中蓝框P3中的3 x rCSP

[-1, 1, Conv, [256, 3, 1]], # 22 对应图中灰框P3的conv k=3,s=1

[-2, 1, Conv, [256, 3, 2]], # 23 对应图中黄框P4中的conv k=3,s=2
[[-1, 16], 1, Concat, [1]], # 24 对应图中黄框P4中的concat, 将蓝框P4和上面的特征融
合
[-1, 3, BottleneckCSP2, [256]], # 25 对应图中黄框P4中的3 x rCSP

[-1, 1, Conv, [512, 3, 1]], # 26 对应图中灰框P4的conv k=3,s=1

[-2, 1, Conv, [512, 3, 2]], # 27 对应图中黄框P5中的conv k=3,s=2
[[-1, 11], 1, Concat, [1]], # 28 对应图中黄框P5中的concat, 将蓝框P5和上面的特征融
合
[-1, 3, BottleneckCSP2, [512]], # 29 对应图中黄框P5中的3 x rCSP

[-1, 1, Conv, [1024, 3, 1]], # 30 对应图中灰框P5的conv k=3,s=1

##在网络结构list中的index为22、26、30的layer
[[22,26,30], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5) 对应图中灰色框
]

```

- 模型主文件: models/yolo.py

```

from n      params  module
arguments
0          -1  1      928  models.common.Conv      [3,
32, 3, 1]

```

1	-1	1	18560	models.common.Conv	[32,
64, 3, 2]					
2	-1	1	19904	models.common.BottleneckCSP	[64,
64, 1]					
3	-1	1	73984	models.common.Conv	[64,
128, 3, 2]					
4	-1	1	161152	models.common.BottleneckCSP	[128,
128, 3]					
5	-1	1	295424	models.common.Conv	[128,
256, 3, 2]					
6	-1	1	2614016	models.common.BottleneckCSP	[256,
256, 15]					
7	-1	1	1180672	models.common.Conv	[256,
512, 3, 2]					
8	-1	1	10438144	models.common.BottleneckCSP	[512,
512, 15]					
9	-1	1	4720640	models.common.Conv	[512,
1024, 3, 2]					
10	-1	1	20728832	models.common.BottleneckCSP	
[1024, 1024, 7]					
11	-1	1	7610368	models.common.SPPCSP	
[1024, 512, 1]					
12	-1	1	131584	models.common.Conv	[512,
256, 1, 1]					
13	-1	1	0	torch.nn.modules.upsampling.Upsample	
[None, 2, 'nearest']					
14	8	1	131584	models.common.Conv	[512,
256, 1, 1]					
15	[-1, -2]	1	0	models.common.Concat	[1]
16	-1	1	2298880	models.common.BottleneckCSP2	[512,
256, 3]					
17	-1	1	33024	models.common.Conv	[256,
128, 1, 1]					
18	-1	1	0	torch.nn.modules.upsampling.Upsample	
[None, 2, 'nearest']					
19	6	1	33024	models.common.Conv	[256,
128, 1, 1]					
20	[-1, -2]	1	0	models.common.Concat	[1]
21	-1	1	576000	models.common.BottleneckCSP2	[256,
128, 3]					
22	-1	1	295424	models.common.Conv	[128,
256, 3, 1]					
23	-2	1	295424	models.common.Conv	[128,
256, 3, 2]					
24	[-1, 16]	1	0	models.common.Concat	[1]
25	-1	1	2298880	models.common.BottleneckCSP2	[512,
256, 3]					
26	-1	1	1180672	models.common.Conv	[256,
512, 3, 1]					
27	-2	1	1180672	models.common.Conv	[256,
512, 3, 2]					
28	[-1, 11]	1	0	models.common.Concat	[1]
29	-1	1	9185280	models.common.BottleneckCSP2	
[1024, 512, 3]					

```

30          -1 1 4720640 models.common.Conv [512,
1024, 3, 1]
31 [22, 26, 30] 1 43080 Detect [1,
[[13, 17, 31, 25, 24, 51, 61, 45], [48, 102, 119, 96, 97, 189, 217, 184], [171,
384, 324, 451, 616, 618, 800, 800]], [256, 512, 1024]]
Model Summary: 476 layers, 7.02668e+07 parameters, 7.02668e+07 gradients

Model(
  (model): Sequential(
    (0): Conv(
      (conv): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn): BatchNorm2d(32, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (1): Conv(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
      (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (2): BottleneckCSP(
      (cv1): Conv(
        (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(32, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (cv3): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (cv4): Conv(
        (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
      (m): Sequential(
        (0): Bottleneck(
          (cv1): Conv(
            (conv): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (bn): BatchNorm2d(32, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
          )
          (cv2): Conv(
            (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
            (bn): BatchNorm2d(32, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
          )
        )
      )
    )
  )
)

```

```

    )
    )
    (3): Conv(
      (conv): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
      (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (4): BottleneckCSP(
      (cv1): Conv(
        (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (cv3): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (cv4): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
      (m): Sequential(
        (0): Bottleneck(
          (cv1): Conv(
            (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
          )
          (cv2): Conv(
            (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
            (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
          )
        )
      )
      (1): Bottleneck(
        (cv1): Conv(
          (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
          (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
        (cv2): Conv(
          (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
          (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
      )
    )
  )

```

```

    )
    )
    (2): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn): BatchNorm2d(64, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
  )
  (5): Conv(
    (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (6): BottleneckCSP(
    (cv1): Conv(
      (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (cv2): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (cv3): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (cv4): Conv(
      (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
    (m): Sequential(
      (0): Bottleneck(
        (cv1): Conv(
          (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
          (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
        (cv2): Conv(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)

```

```

        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(1): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(
    (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
)
(2): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(
    (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
)
(3): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(
    (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
)
(4): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```



```

        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(5): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(6): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(7): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)

```

```

    )
    )
    (8): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (9): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (10): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (11): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )

```

```

        (cv2): Conv(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
          (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
      )
    (12): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (13): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (14): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
  )
)
)
)

```

```

(7): Conv(
  (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
  (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
  (act): MishCuda()
)
(8): BottleneckCSP(
  (cv1): Conv(
    (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (cv3): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (cv4): Conv(
    (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
  (act): MishCuda()
  (m): Sequential(
    (0): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (1): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
  )
)

```

```

(2): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(
    (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
)
(3): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(
    (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
)
(4): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(
    (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
)
(5): Bottleneck(
  (cv1): Conv(
    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv(

```

```

        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(6): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(7): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(8): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(9): Bottleneck(
    (cv1): Conv(

```

```

        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(10): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(11): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(12): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)

```

```

        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(13): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(14): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
)
(9): Conv(
    (conv): Conv2d(512, 1024, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
    (bn): BatchNorm2d(1024, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
)
(10): BottleneckCSP(
    (cv1): Conv(
        (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (cv3): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (cv4): Conv(
        (conv): Conv2d(1024, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```



```

        (bn): BatchNorm2d(1024, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (bn): BatchNorm2d(1024, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
    (m): Sequential(
      (0): Bottleneck(
        (cv1): Conv(
          (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
          (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
        (cv2): Conv(
          (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
          (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
      )
    (1): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (2): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (3): Bottleneck(
      (cv1): Conv(

```

```

        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(4): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(5): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(6): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)

```

```

        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
)
(11): SPPCSP(
    (cv1): Conv(
        (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (cv3): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv4): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (m): ModuleList(
        (0): MaxPool2d(kernel_size=5, stride=1, padding=2, dilation=1,
ceil_mode=False)
        (1): MaxPool2d(kernel_size=9, stride=1, padding=4, dilation=1,
ceil_mode=False)
        (2): MaxPool2d(kernel_size=13, stride=1, padding=6, dilation=1,
ceil_mode=False)
    )
    (cv5): Conv(
        (conv): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv6): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (bn): BatchNorm2d(1024, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
    (cv7): Conv(
        (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)

```

```

    )
    (12): Conv(
      (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (13): Upsample(scale_factor=2.0, mode=nearest)
    (14): Conv(
      (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
    )
    (15): Concat()
    (16): BottleneckCSP2(
      (cv1): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (cv3): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
      (act): MishCuda()
      (m): Sequential(
        (0): Bottleneck(
          (cv1): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
          )
          (cv2): Conv(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
            (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
          )
        )
      )
      (1): Bottleneck(
        (cv1): Conv(
          (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
          (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
        (cv2): Conv(

```

```

        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
(2): Bottleneck(
    (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
)
)
(17): Conv(
    (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
)
(18): Upsample(scale_factor=2.0, mode=nearest)
(19): Conv(
    (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
)
(20): Concat()
(21): BottleneckCSP2(
    (cv1): Conv(
        (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (cv2): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (cv3): Conv(
        (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
    (m): Sequential(
        (0): Bottleneck(

```

```

        (cv1): Conv(
          (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
          (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
        (cv2): Conv(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
          (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
      )
    (1): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (2): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(128, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
  )
  (22): Conv(
    (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (23): Conv(

```

```

        (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
    )
    (24): Concat()
    (25): BottleneckCSP2(
        (cv1): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
        )
        (cv2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (cv3): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
            (act): MishCuda()
        )
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
        (m): Sequential(
            (0): Bottleneck(
                (cv1): Conv(
                    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
                    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
                    (act): MishCuda()
                )
                (cv2): Conv(
                    (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
                    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
                    (act): MishCuda()
                )
            )
            (1): Bottleneck(
                (cv1): Conv(
                    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
                    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
                    (act): MishCuda()
                )
                (cv2): Conv(
                    (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
                    (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
                    (act): MishCuda()
                )
            )
            (2): Bottleneck(

```

```

        (cv1): Conv(
          (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
          (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
        (cv2): Conv(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
          (bn): BatchNorm2d(256, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
      )
    )
  )
(26): Conv(
  (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
  (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
  (act): MishCuda()
)
(27): Conv(
  (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
  (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
  (act): MishCuda()
)
(28): Concat()
(29): BottleneckCSP2(
  (cv1): Conv(
    (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (cv2): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (cv3): Conv(
    (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
    (act): MishCuda()
  )
  (bn): BatchNorm2d(1024, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
  (act): MishCuda()
  (m): Sequential(
    (0): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )

```



```

        (cv2): Conv(
          (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
          (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
          (act): MishCuda()
        )
      )
    (1): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
    (2): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
      (cv2): Conv(
        (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
        (bn): BatchNorm2d(512, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
        (act): MishCuda()
      )
    )
  )
)
)
(30): Conv(
  (conv): Conv2d(512, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
  (bn): BatchNorm2d(1024, eps=0.001, momentum=0.03, affine=True,
track_running_stats=True)
  (act): MishCuda()
)
(31): Detect(
  (m): ModuleList(
    (0): Conv2d(256, 24, kernel_size=(1, 1), stride=(1, 1))
    (1): Conv2d(512, 24, kernel_size=(1, 1), stride=(1, 1))
    (2): Conv2d(1024, 24, kernel_size=(1, 1), stride=(1, 1))
  )
)
)
)
)

```

2. Train Custom Data

2.1 Create dataset.yaml

```
# train and val datasets (image directory or *.txt file with image paths)
train: ../person/images/train # 118k images
val: ../person/images/val # 5k images

# number of classes
nc: 1

# class names
names: ['person']
```

2.2 Create Labels

one `*.txt` file per image (if no objects in image, no `*.txt` file is required). The `*.txt` file specifications are:

- One row per object
- Each row is `class x_center y_center width height` format.
- Box coordinates must be in **normalized xywh** format (from 0 - 1). If your boxes are in pixels, divide `x_center` and `width` by image width, and `y_center` and `height` by image height.
- Class numbers are zero-indexed (start from 0).

2.3 Organize Directories

```
/person
  /images
    /train
      001.jpg
    /val
      007.jpg
/person
  /labels
    /train
      001.txt
    /val
      007.txt
```

2.4 修改模型配置文件

- 以yolov4-p5.yaml为例:

```
# parameters
nc: 1 # number of classes
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple
```

注意，其实不修改也行，因为作者在代码中进行了处理：

```
##opt.data, 数据文件(coco.yaml)
with open(opt.data) as f:
    data_dict = yaml.load(f, Loader=yaml.FullLoader) # model dict
    train_path = data_dict['train']
    test_path = data_dict['val']
    ##nc是以数据文件(coco.yaml)中的nc为准，在后续处理中会将模型结构(例如yolov4-p5.yaml)中的nc设置为数据文件(coco.yaml)中的nc
    nc, names = (1, ['item']) if opt.single_cls else (int(data_dict['nc']),
data_dict['names']) # number classes, names
    assert len(names) == nc, '%g names found for nc=%g dataset in %s' %
(len(names), nc, opt.data) # check
```

```
model = Model(opt.cfg or ckpt['model'].yaml, ch=3, nc=nc).to(device) # create
```

```
##模型主类
class Model(nn.Module):
    def __init__(self, cfg='yolov4-p5.yaml', ch=3, nc=None): # model, input
channels, number of classes
        super(Model, self).__init__()
        # Define model
        if nc and nc != self.yaml['nc']:
            print('Overriding %s nc=%g with nc=%g' % (cfg, self.yaml['nc'], nc))
            ##nc是以数据文件(coco.yaml)中的nc为准，在这里将模型结构(例如yolov4-p5.yaml)
中的nc设置为数据文件(coco.yaml)中的nc
            self.yaml['nc'] = nc # override yaml value
```

问雪更新于20210113