

姿勢推定に向けたセグメンテーションモデル構築状況報告

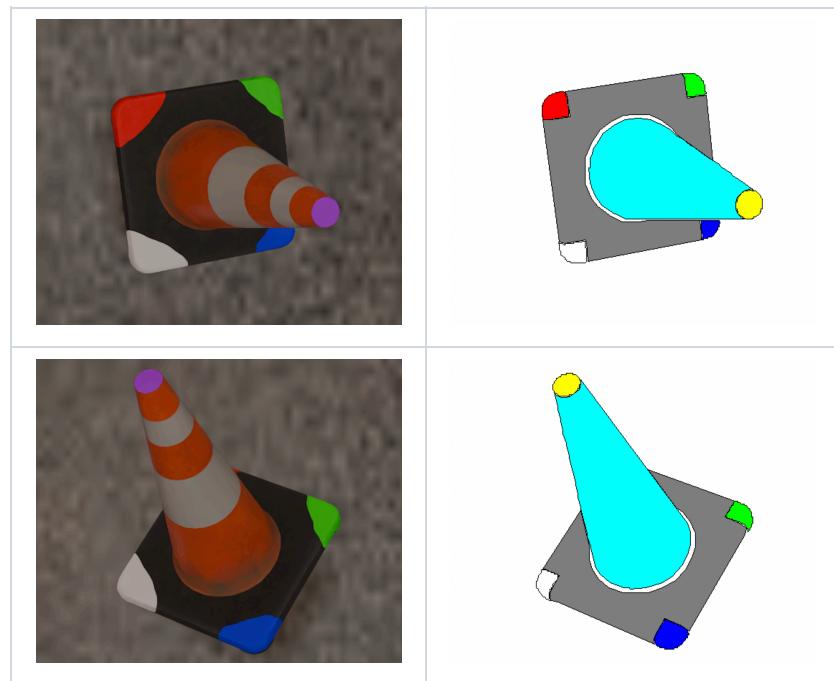
コーンのセグメンテーション完了

0. 前回まで

サイドスラスターを「コーン」に見立て、姿勢推進のためのセグメンテーションモデルの構築を開始。

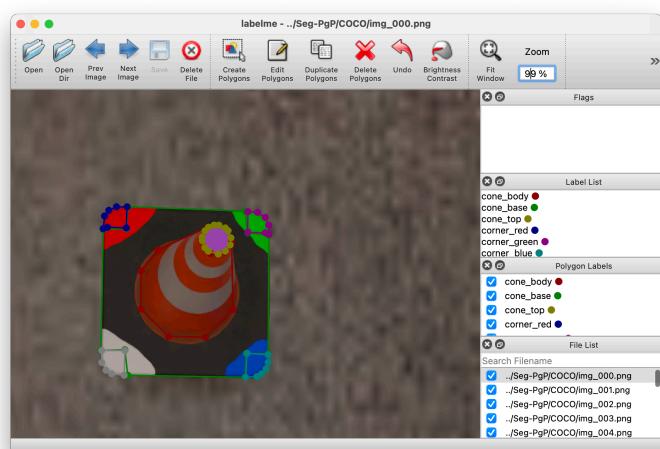
1. 学習データ見直し

400枚のデータで学習。



目視確認&修正

数回のデータ修正を目視確認を経て実施。



2. 最終結果

```
30/99      0G   0.03691   0.0203   0.03183   0.007395    76   1024: 100% |  
██████████| 40/40 [07:42<00:00, 11.56s/it]  
Class     Images Instances Box(P)      R       mAP50  mAP50-95) Mask(P)      R  
mAP50  mAP50-95): 100% |██████████|  
all      80      537    0.901    0.958    0.975    0.645    0.901    0.958  
0.972    0.766
```

項目	値
Box mAP50	0.975
Box mAP50-95	0.645
Mask mAP50	0.972
Mask mAP50-95	0.766
Precision	0.901
Recall	0.958
学習データ枚数	80 images
インスタンス数	537
エポック	30 / 100

⚠ 注意：最終的なデータ 今回の学習データは、コーンの位置は

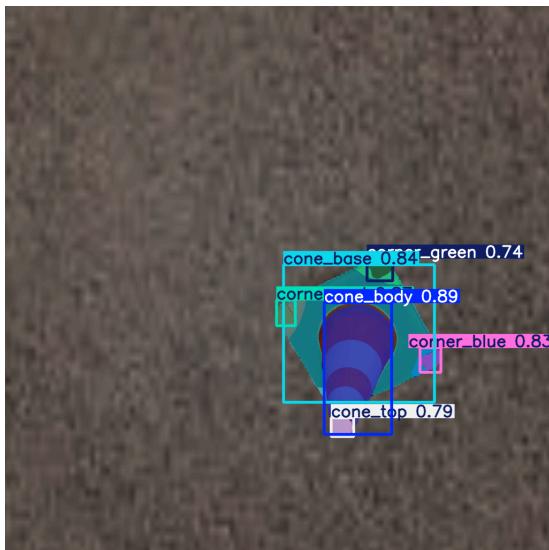
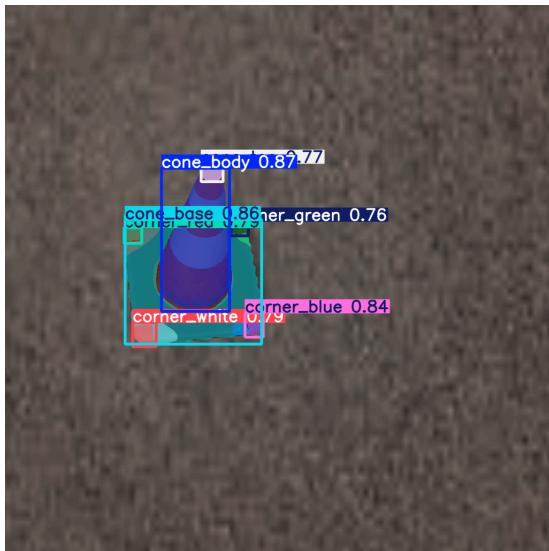
- $x=\pm 0.4\text{m}$, $y=\pm 0.4\text{m}$, $z=2.0\text{m}$
- $(\text{yaw}, \text{pitch}, \text{roll})=(\pm 30^\circ, \pm 30^\circ, \pm 30^\circ)$

の範囲でランダム。

最終的な学習データは、サイドスラスターの画像を使用して、広範囲の位置や向きでデータ作成。2,000枚程度の学習データにより、上記と同等の精度を期待できる。

3. 推論結果

80枚を推論（コーンのセグメントを予測）した結果、概ね以下のように良好な結果：



4. s と n モデル比較

Raspberry Pi 稼働を考慮して（小さいモデルほど良い）、2.0 MBの YOLOv5n-seg を YOLOv5s-seg と同条件で学習。

Model	size (pixels)	mAP ^{box} 50-95	mAP ^{mask} 50-95	Train time 300 epochs A100 (hours)	Speed ONNX CPU (ms)	Speed TRT A100 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n-seg	640	27.6	23.4	80:17	62.7	1.2	2.0	7.1
YOLOv5s-seg	640	37.6	31.7	88:16	173.3	1.4	7.6	26.4
YOLOv5m-seg	640	45.0	37.1	108:36	427.0	2.2	22.0	70.8
YOLOv5l-seg	640	49.0	39.9	66:43 (2x)	857.4	2.9	47.9	147.7
YOLOv5x-seg	640	50.7	41.4	62:56 (3x)	1579.2	4.5	88.8	265.7

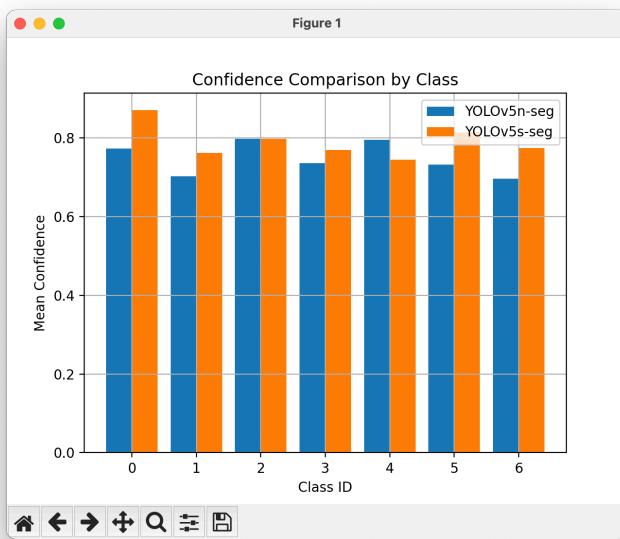
Confidenceとは

各セグメントに対して「これは正しい検出」と判断した“自信度”（confidence）を0~1で出力。これは統計的な意味での確率ではなく、**モデルがどれだけ強くそう判断しているか**を示すスコア（学習データ分布の範囲内での“主観的な確信”、とも言える）。「足切り」（閾値）として主に利用。

以下の s と n のモデル confidence 比較において、両者の精度に大きな違いはない。

n モデルはこの後も構築対象とすべき。最終的な選択は、**姿勢推定の精度比較**で決定。

```
class_id      conf_n      conf_s
0            0.773521  0.870380
1            0.703173  0.761920
2            0.797858  0.798302
3            0.735879  0.769904
4            0.795480  0.744254
5            0.732295  0.813823
6            0.697050  0.774604
```



YOLOv5s-seg

	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	361984	models.common.C3	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90880	models.common.C3	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	296448	models.common.C3	[256, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C3	[512, 512, 1, False]
24	[17, 20, 23]	1	418252	models.yolo.Segment	[7, [[10, 13, 16, 30,
33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], 32, 128, [128, 256, 512]]					

...

YOLOv5n-seg

...

	from	n	params	module	arguments
0	-1	1	1760	models.common.Conv	[3, 16, 6, 2, 2]
1	-1	1	4672	models.common.Conv	[16, 32, 3, 2]
2	-1	1	4800	models.common.C3	[32, 32, 1]
3	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
4	-1	2	29184	models.common.C3	[64, 64, 2]
5	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
6	-1	3	156928	models.common.C3	[128, 128, 3]
7	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
8	-1	1	296448	models.common.C3	[256, 256, 1]
9	-1	1	164608	models.common.SPPF	[256, 256, 5]
10	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	90880	models.common.C3	[256, 128, 1, False]
14	-1	1	8320	models.common.Conv	[128, 64, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	22912	models.common.C3	[128, 64, 1, False]
18	-1	1	36992	models.common.Conv	[64, 64, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	74496	models.common.C3	[128, 128, 1, False]
21	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	296448	models.common.C3	[256, 256, 1, False]
24	[17, 20, 23]	1	135628	models.yolo.Segment	[7, [[10, 13, 16, 30,
33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], 32, 64, [64, 128, 256]]					

...

```
```
khi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
khi@raspberrypi:~ $ uname -a
Linux raspberrypi 6.12.34+rpt-rpi-2712 #1 SMP PREEMPT Debian 1:6.12.34-1+rpt1~bookworm (2025-06-
26) aarch64 GNU/Linux
```
```