

Raspberry Pi でのセグメント検知

ラップトップ（M3 Mac）で開発したセグメンテーションモデルを Raspberry Pi5 で稼働するように「重みファイル変換」。具体的には、ONNX (Open Neural Network Exchange) 形式に変換。

手法： 二つのONNX変換（FP32, FP16）を作成して、400枚の画像のセグメント検知の精度・速度を比較。

Raspberry Pi5

55.0度を超えたらファンが起動した。

2 秒毎に温度表示：

```
Every 2.0s: vcgencmd measure_temp          raspberrypi: Sun Aug 17 09:08:40 2025  
  
temp=59.3'C
```

最終的に60度を超えることはなかった。ただし、**ケース加納していない**ため、格納時はより高温になると予想。

80 度前後で性能低下 サーマルスロットリング（Thermal Throttling）により性能低下。

結果

=== FP16 ===
Model: fastseg_fp16.onnx (tensor(float16) -> tensor(float16))
Images: 400
Speed : avg=681.5 ms | med=681.6 | p95=685.3
Quality: mIoU=0.8466 | pixAcc=0.9970
Per-class IoU / Recall:
ID0: IoU=0.9979 Recall=0.9988
ID1: IoU=0.9561 Recall=0.9717
ID2: IoU=0.8250 Recall=0.8707
ID3: IoU=0.9325 Recall=0.9816
ID4: IoU=0.7487 Recall=0.7768
ID5: IoU=0.7887 Recall=0.8427
ID6: IoU=0.7774 Recall=0.8373
ID7: IoU=0.7468 Recall=0.7867

=== FP32 ===
Model: fastseg_fp32.onnx (tensor(float) -> tensor(float))
Images: 400
Speed : avg=868.7 ms | med=868.8 | p95=871.8
Quality: mIoU=0.8459 | pixAcc=0.9970
Per-class IoU / Recall:
ID0: IoU=0.9979 Recall=0.9988
ID1: IoU=0.9563 Recall=0.9718
ID2: IoU=0.8250 Recall=0.8709
ID3: IoU=0.9323 Recall=0.9817
ID4: IoU=0.7462 Recall=0.7730
ID5: IoU=0.7873 Recall=0.8401
ID6: IoU=0.7765 Recall=0.8352
ID7: IoU=0.7461 Recall=0.7858

=== SUMMARY (FP16 vs FP32) ===
Speed avg(ms): 681.5 vs 868.7
mIoU : 0.8466 vs 0.8459
pixAcc : 0.9970 vs 0.9970

項目	FP16	FP32	結果
速度 (ms)	681.5	868.7	FP16が約 21.5% 高速
精度 (mIoU)	0.8466	0.8459	ほぼ同じ
精度 (pixAcc)	0.9970	0.9970	同じ

mIoU (mean Intersection over Union) クラスごとの IoU を平均、クラス間のバランスを評価。セグメンテーションの代表的な指標。

pixAcc (Pixel Accuracy) 全ピクセルのうち、正しく分類されたピクセルの割合。全体的な「正解率」で、**背景が多いデータだと高めに出やすい**。

結論

背景エリアが広いため pixAcc は高く出ている。ただし小さいクラスの IoU も全て 0.5 以上を確保しており、実用上十分な精度と判断できる。

M3 MacBookAirの場合

```
=== FP16 ===
Model: fastseg_fp16.onnx (tensor(float16) -> tensor(float16))
Images: 400
Speed  : avg=153.8 ms | med=153.1 | p95=158.9
Quality: mIoU=0.8458 | pixAcc=0.9970
Per-class IoU / Recall:
  ID0: IoU=0.9979  Recall=0.9988
  ID1: IoU=0.9562  Recall=0.9717
  ID2: IoU=0.8249  Recall=0.8705
  ID3: IoU=0.9323  Recall=0.9816
  ID4: IoU=0.7449  Recall=0.7714
  ID5: IoU=0.7880  Recall=0.8418
  ID6: IoU=0.7766  Recall=0.8355
  ID7: IoU=0.7461  Recall=0.7863
```

```
=== FP32 ===
Model: fastseg_fp32.onnx (tensor(float) -> tensor(float))
Images: 400
Speed  : avg=157.8 ms | med=156.9 | p95=166.2
Quality: mIoU=0.8459 | pixAcc=0.9970
Per-class IoU / Recall:
  ID0: IoU=0.9979  Recall=0.9988
  ID1: IoU=0.9563  Recall=0.9718
  ID2: IoU=0.8250  Recall=0.8709
  ID3: IoU=0.9323  Recall=0.9817
  ID4: IoU=0.7462  Recall=0.7730
  ID5: IoU=0.7873  Recall=0.8401
  ID6: IoU=0.7765  Recall=0.8352
  ID7: IoU=0.7461  Recall=0.7858
```

```
=== SUMMARY (FP16 vs FP32) ===
Speed avg(ms): 153.8  vs  157.8
mIoU          : 0.8458 vs 0.8459
pixAcc        : 0.9970 vs 0.9970
```

項目	FP16	FP32	結果
速度 (ms)	153.8	157.8	ほぼ同じ
精度 (mIoU)	0.8458	0.8459	ほぼ同じ
精度 (pixAcc)	0.9970	0.9970	同じ

サイドスラスター Blender 変換

元ファイル（STP）からの変換で色が落ちるため、実物と色を合わせる必要あり。

質問：サイズは 0.86 m × 0.67 m × 0.86 m（直径 86cm、高さ 67cm）か？

注意：会社事情により以下の画像表示は不可。

