

第4回AIエッジコンテスト (実装コンテスト②) 実装レポート

チーム名：ymym

自己紹介

松尾康弘 (ymym)

■仕事

日系メーカーでカメラ開発 (2016/4～)

■参加のきっかけ

2015年

- ・ Microsoft BingのFPGA活用事例を見てFPGAによる高速化に興味を持つ

↓勉強会などに参加したが、ハードルが高く、モノを作ることができず

2020年

- ・ AIエッジコンテストが開催されていることを知る
- ・ 5年前と比べて、FPGA、Deep Learningのツールが整備された

興味があったFPGAによるDeep Learningの高速化に挑戦！

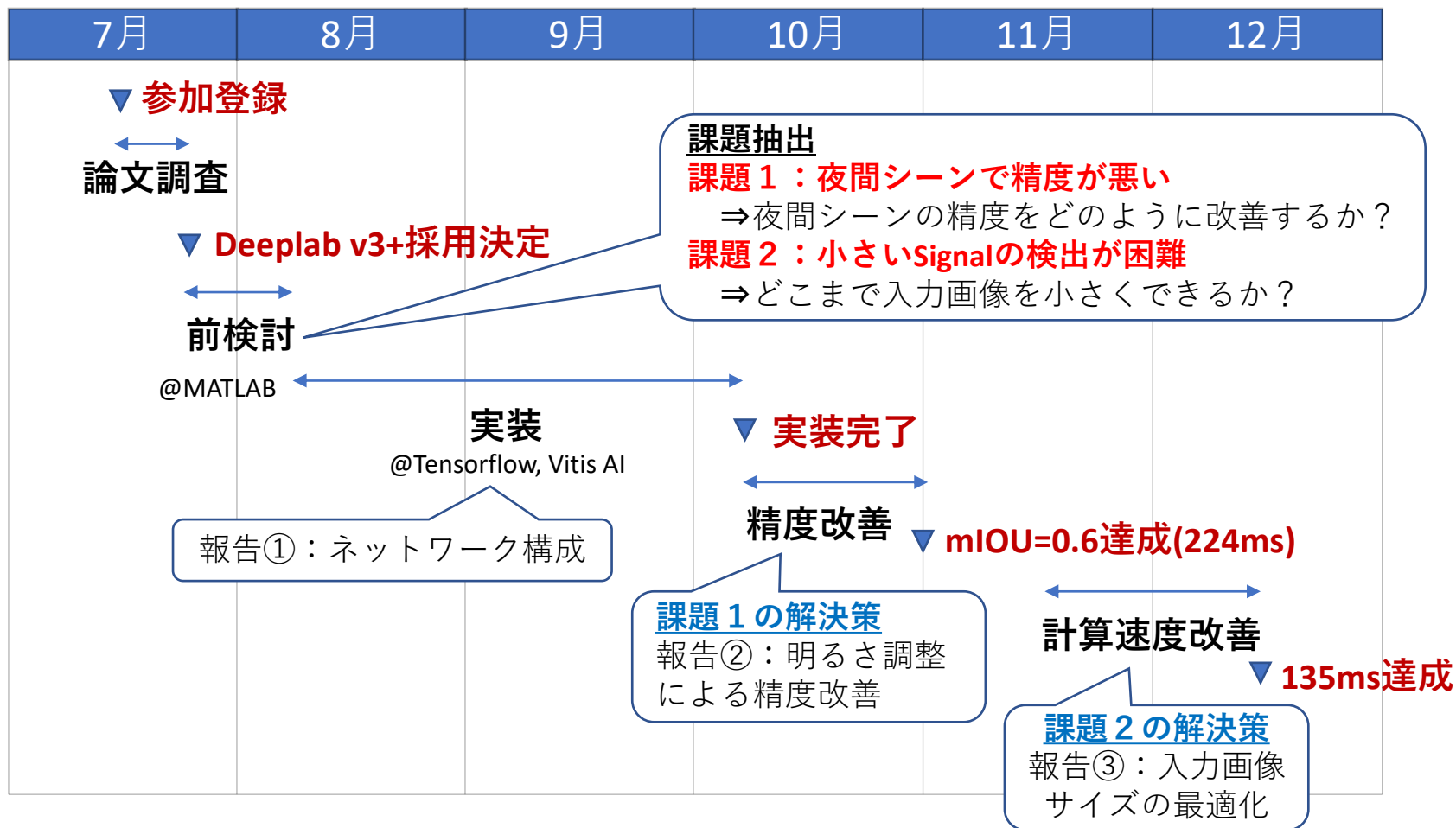
報告内容

最終結果

- ・ 使用ネットワーク : DeepLab v3+
- ・ 実装 : VitisAIで回路化(DPU:B2304)、 argmaxのみCPU処理
- ・ スコア : **mIOU=0.6005591**
- ・ 処理時間 : **135.46msec**

- ・ 開発日程
- ・ 工夫したポイント
 - ①ネットワーク構成
 - ②明るさ調整による精度改善
 - ③入力画像サイズの最適化
- ・ 考察：二分割による弊害

開発日程



トータル作業時間：500～600時間程度

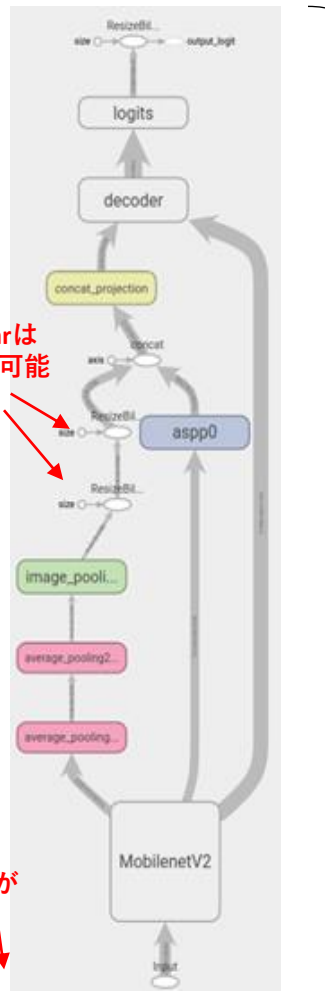
①ネットワーク構成

回路化したネットワーク

出力： $384 \times 384 \times 5$
(argmax処理前)

Signal,
Pedestrian,
Lane,
Car,
その他

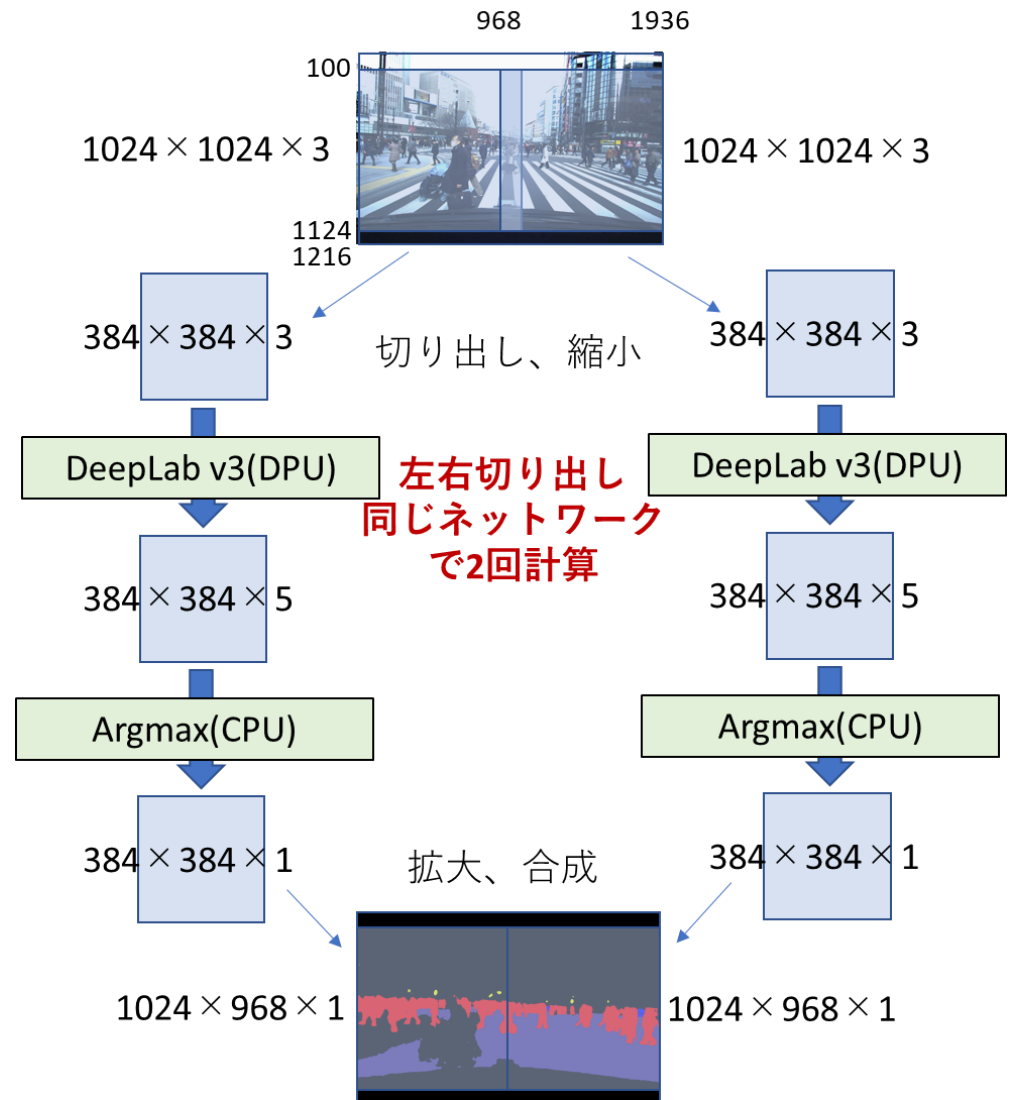
※学習、推論ともに
5カテゴリで実施



Vitis AI
で回路化

入力： $384 \times 384 \times 3$

回路の使いこなし



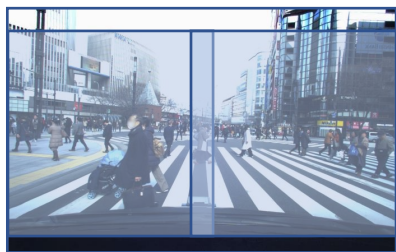
Vitis AIの制約に合わせてネットワークを設計

補足：データオーグメンテーション

推論

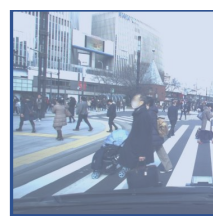
1936

1024



1216

特定領域
をクロップ



1024

スケール
 $\times 0.375$

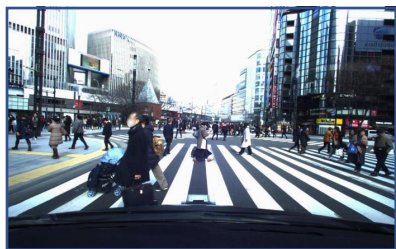
384



384

後述の明るさ調整

学習



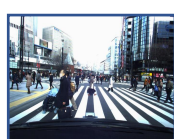
ランダム
スケール

$x0.3 \sim x0.75$
(0.075刻み)

$x0.375$ 基準だと
 $x0.8 \sim x2.0$
(0.2刻み)

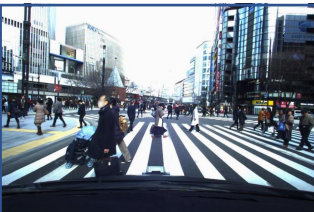


$x0.3$



$x0.375$
(基準)

⋮



$x0.75$

ランダム
クロップ

384×384



左右
フリップ

確率0.5



後述の明るさ調整

**狙い：大きいサイズの
Signal, Pedestrianを多く入れたい**

補足: クロスエントロピー計算の重みも調整 (Signal=6.0, Pedestrian=1.6, 他カテゴリ=0.6, mIOUが0.01程度改善)

1つの学習済みモデルで左右の領域の推論が可能に

課題1：夜間シーンで精度が悪い

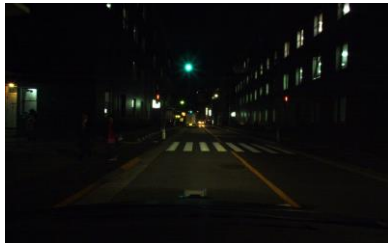
⇒夜間シーンの精度をどのように改善するか？

②明るさ調整による精度改善

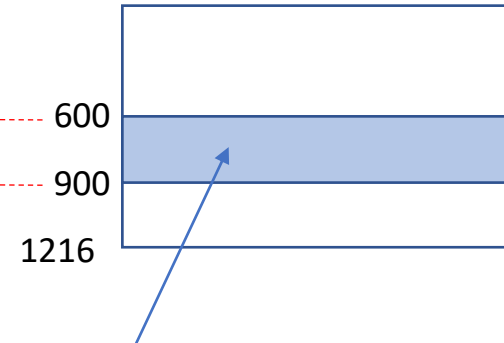
昼夜シーンが混在、明るさを調整したうえで学習、推論を実施

調整前

調整後



1936



遠方地面領域の画素値平均

$$\text{ave} = (\text{ave_R} + \text{ave_G} + \text{ave_B}) / 3$$

128/aveを全画素に掛け算して明るさを調整

狙い

入力画像の正規化処理を適切に行うため

(128を引いて-128～127の範囲にしたい、
低輝度画像はそのままと-128付近に偏る)

mIOU=0.5→0.6に精度が大幅改善

@512×512

課題2：小さいSignalの検出が困難

⇒どこまで入力画像を小さくできるか？

③入力画像サイズの最適化

前処理、後処理、
2回回しのトータル時間

検討過程で
別パラメータも最適化

入力画像サイズ	計算時間	mIOU	学習 パラメータ 最適化	高速化： 明るさ調整、 argmax、 の簡略化
512 × 512	224ms	0.6072		
384 × 384	135ms	0.6006	○	○
256 × 256	84ms	0.5853	○	
128 × 128	38ms	0.5176		

mIOU=0.6を維持しつつ、入力画像を384 × 384まで縮小化、
計算速度改善

考察：二分割による弊害(1/2)

本実装では、領域を二分割して推論した結果を貼り合わせているため、弊害確認

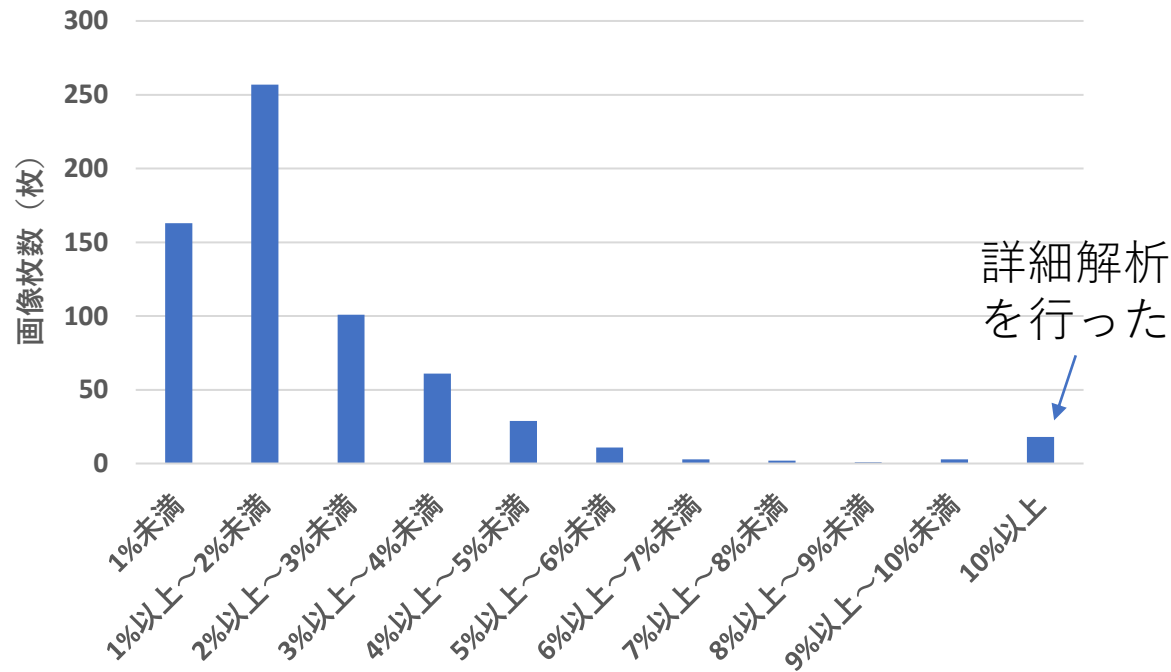
評価方法

オーバーラップ領域において、
左右で結果が異なっている画素の割合（エラー率）を算出

オーバーラップ領域



エラー率の分布



ほとんどの画像において、左右ほぼ同じ結果が得られる
ただし、一部画像では左右で異なる結果が出てしまっている

考察：二分割による弊害(2/2) 下記の3タイプが確認された

タイプ1

車と車以外(バス、トラック、バイク)の判別を間違える (12枚/18枚)

タイプ2

道路を正しく分離できない (4枚/18枚)

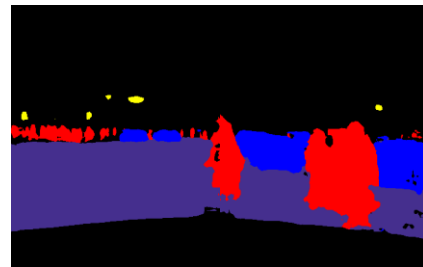
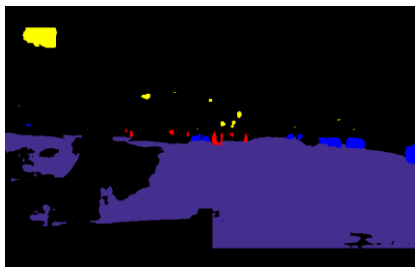
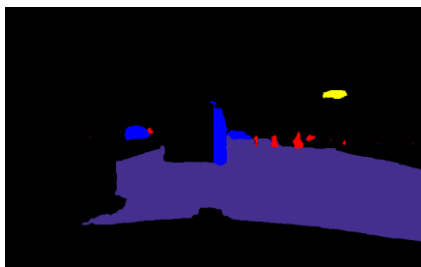
タイプ3

夜間シーンで検出結果が左右でばらつく (2枚/18枚)

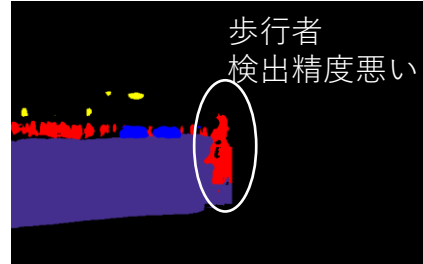
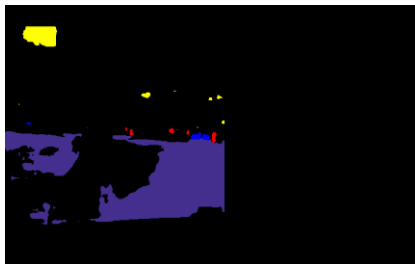
元画像



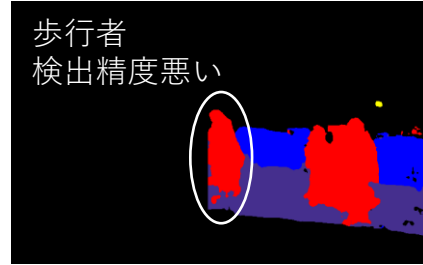
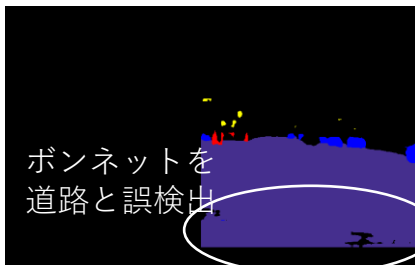
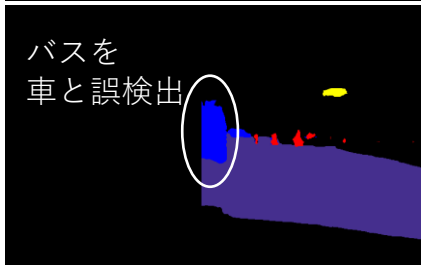
結果
貼り合
わせ後



左結果
貼り合
わせ前



右結果
貼り合
わせ前



まとめ

- ・回路化のために、Vitis AIの制約に合わせてネットワークを設計
- ・ **課題 1 : 夜間シーンで精度が悪い**
⇒明るさ調整による精度改善 (mIOU=0.5 ⇒ 0.6の大幅改善)
- ・ **課題 2 : 小さいSignalの検出が困難**
⇒mIOU=0.6を維持しつつ、入力画像を384×384まで縮小化、計算速度改善



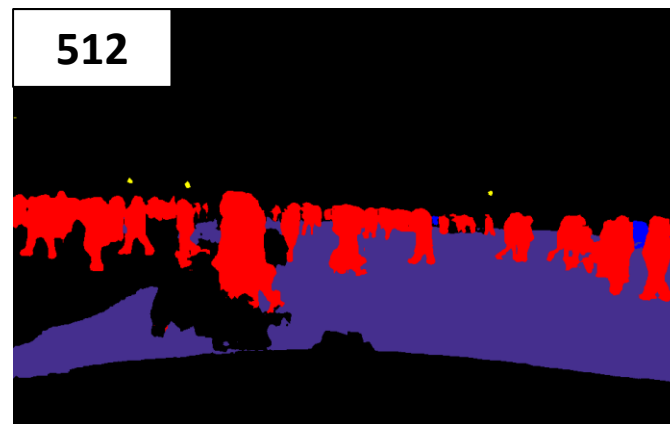
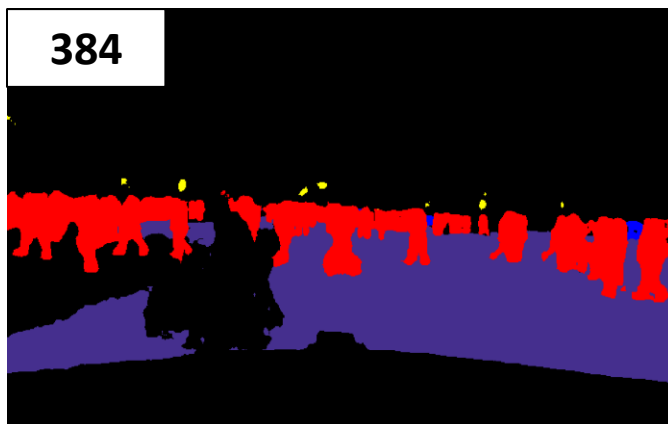
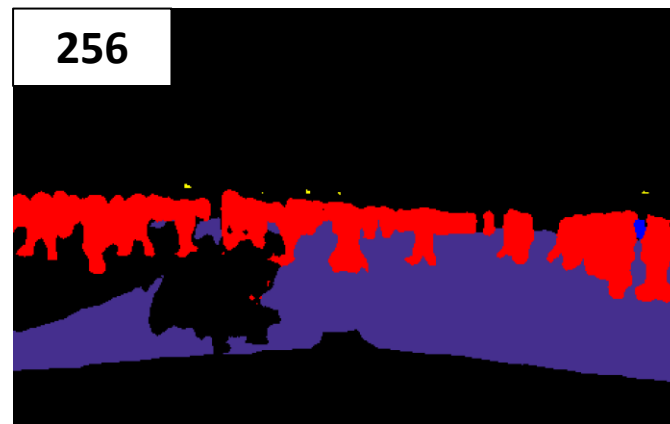
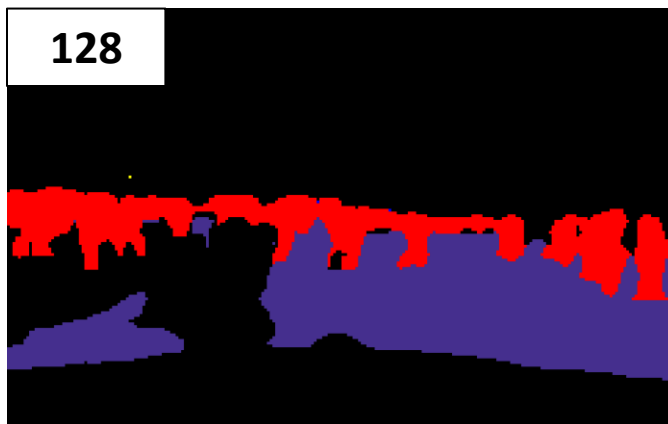
最終結果

- | | |
|------------|---|
| ・ 使用ネットワーク | : DeepLab v3+ |
| ・ 実装 | : VitisAIで回路化(DPU:B2304)、 argmaxのみCPU処理 |
| ・ スコア | : mIOU=0.6005591 |
| ・ 処理時間 | : 135.46msec |

- ・ 二分割による弊害が確認された、今後本実装を活用する場合は検討が必要

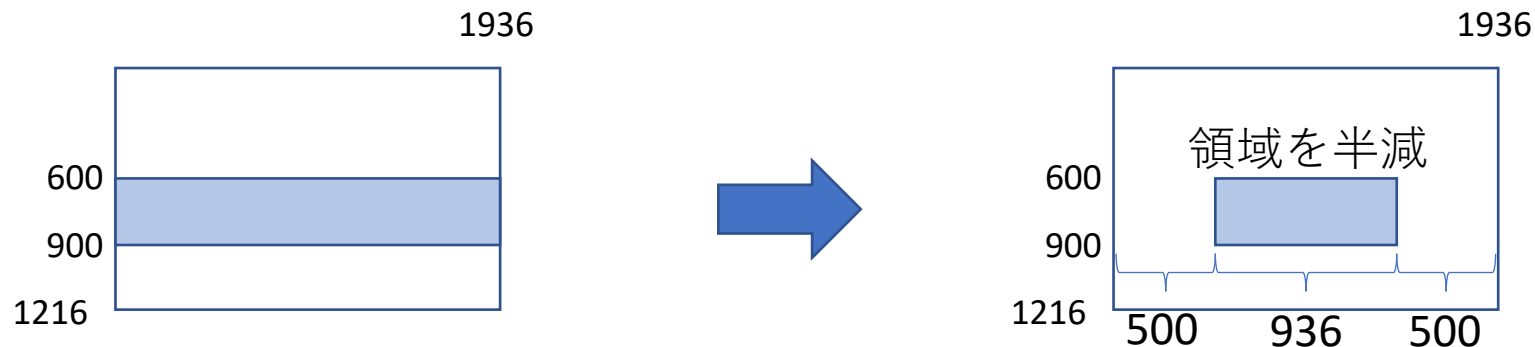
(参考)入力画像サイズ変更時の
セグメンテーション結果

入力画像を小さくすると
セグメンテーション結果も粗くなる



(参考)高速化処理内容

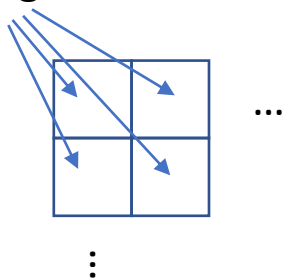
明るさ調整簡略化



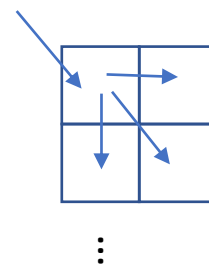
遠方地面領域の画素値平均の計算領域を限定

argmax間引き簡略化

全画素argmax計算



1画素は
argmax計算



隣接3画素は
結果コピー

1画素はargmax計算、隣接3画素は結果コピー

(参考)二分割による弊害の詳細解析

- ・ エラー率**10%**以上の画像**18**枚を詳細解析
- ・ 下記順番で結果表示
 - ・ 元画像
 - ・ セグメンテーション結果（貼り合わせ後）
 - ・ 左セグメンテーション結果（貼り合わせ前）
 - ・ 右セグメンテーション結果（貼り合わせ前）
 - ・ エラー箇所（白がエラー、左右で結果が異なる領域）
- ・ ほぼ同一シーン（信号待ちで同じ位置から撮影）は赤枠で囲っている

test_033

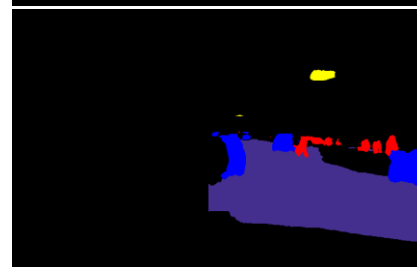
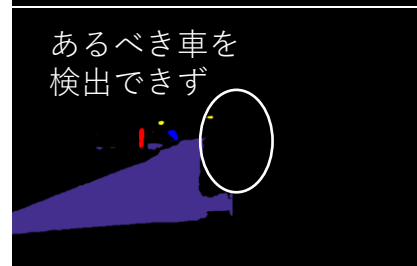
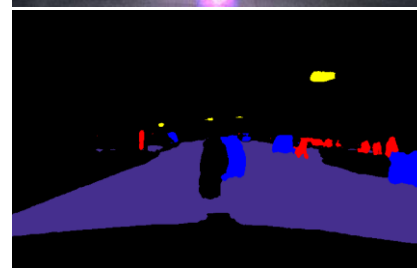
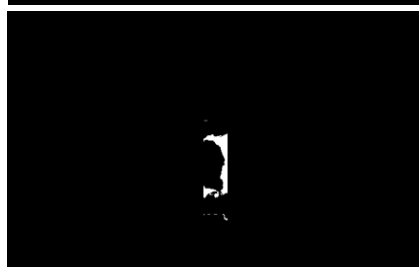
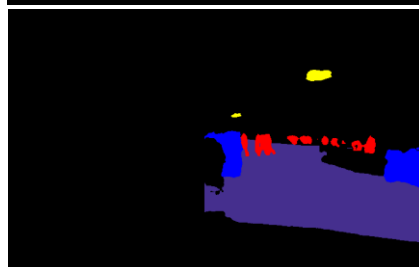
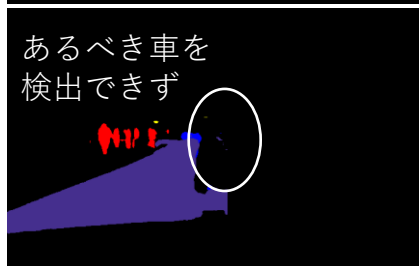
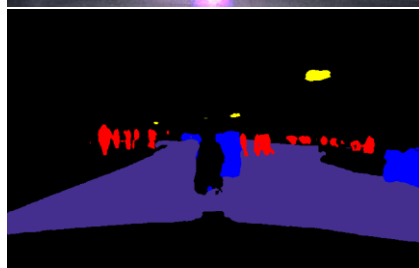
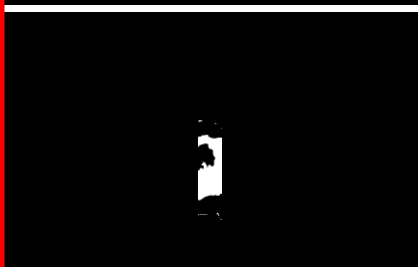
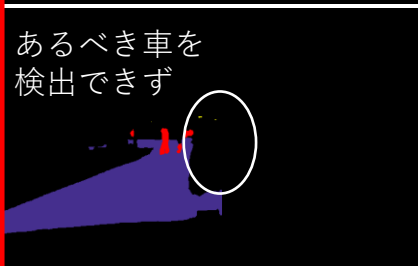
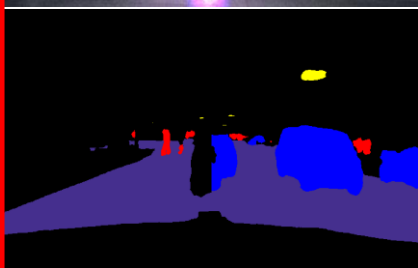
test_078

test_191

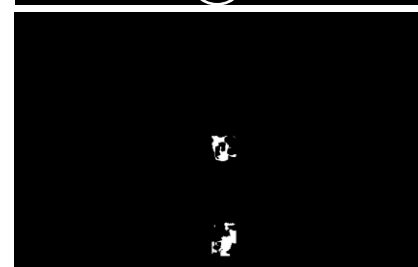
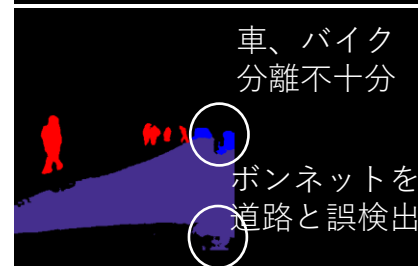
test_123

タイプ1

元画像

結果
貼り合
わせ後左結果
貼り合
わせ前右結果
貼り合
わせ前エラー
箇所

ほぼ同一条件



test_263

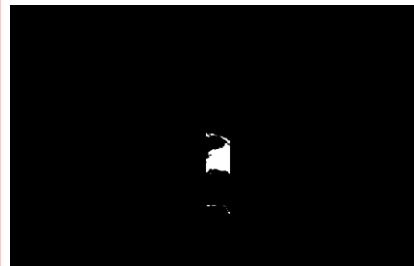
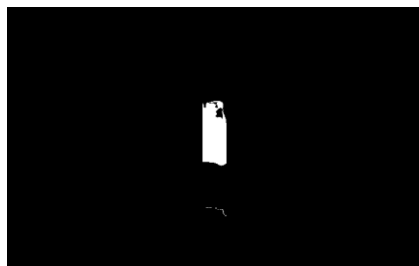
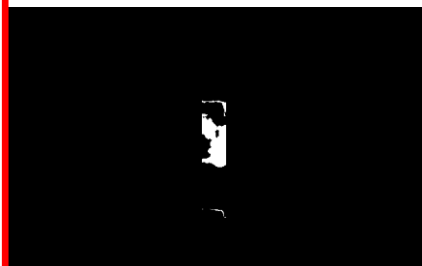
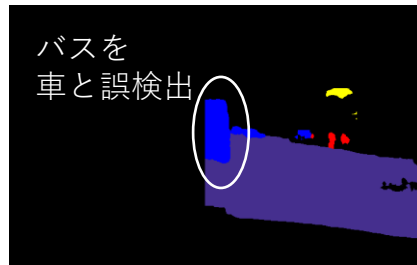
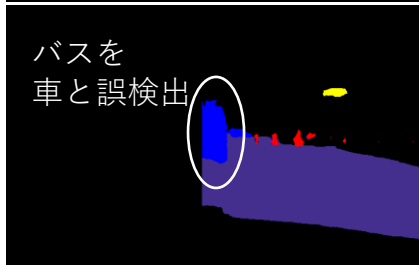
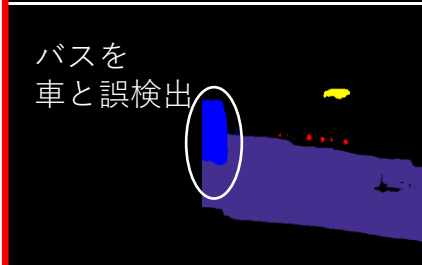
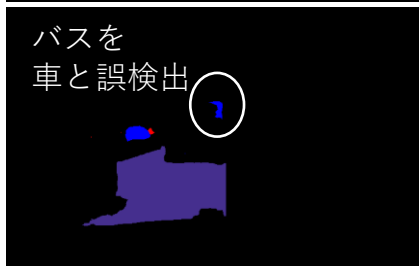
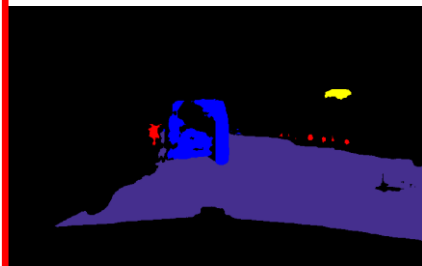
test_292

test_373

test_052

タイプ1

元画像

結果
貼り合
わせ後左結果
貼り合
わせ前右結果
貼り合
わせ前エラー
箇所

ほぼ同一条件

test_337

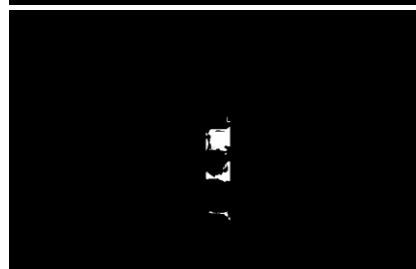
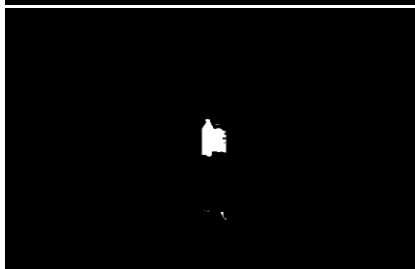
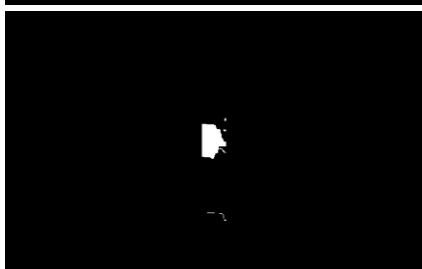
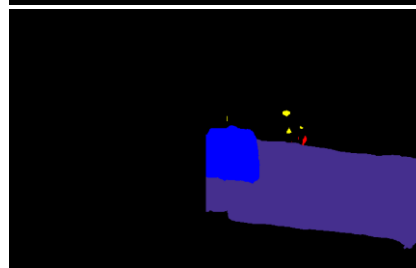
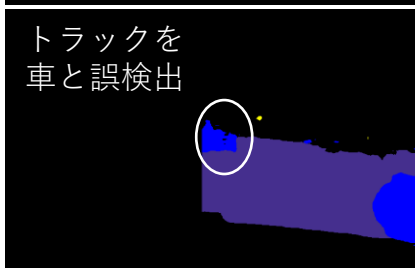
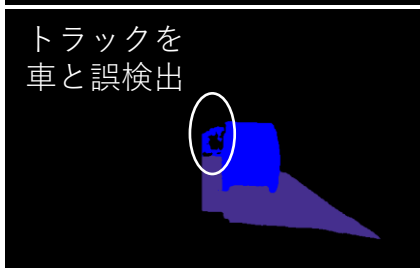
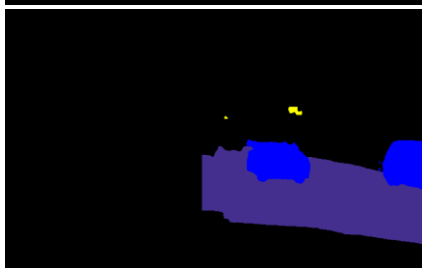
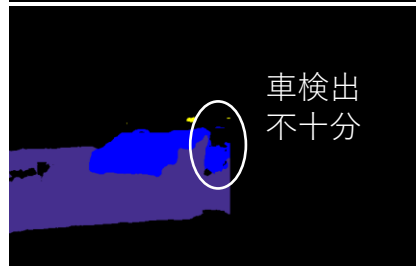
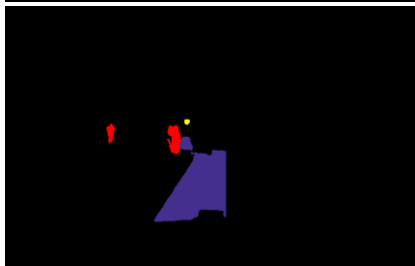
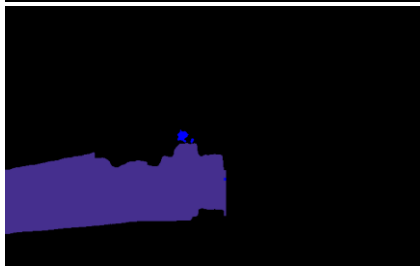
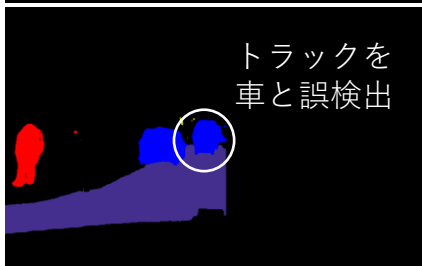
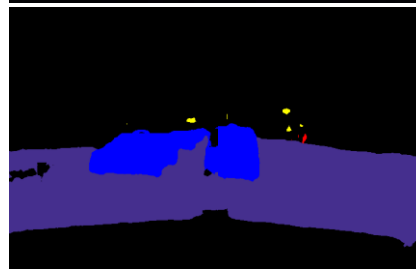
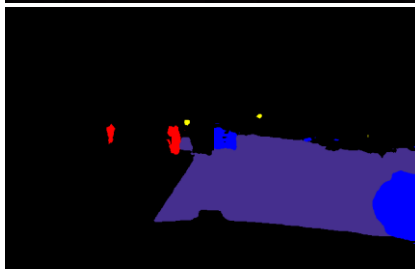
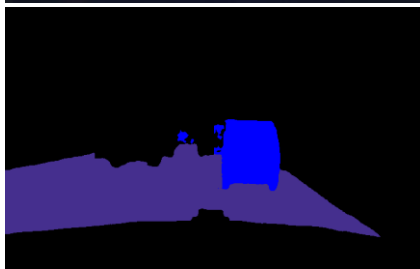
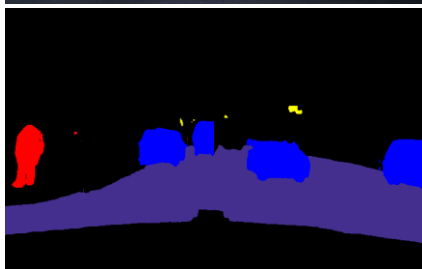
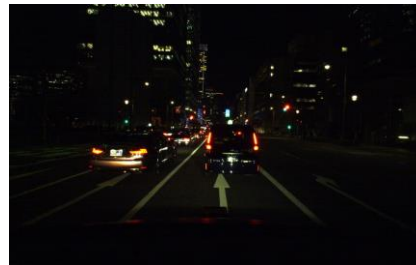
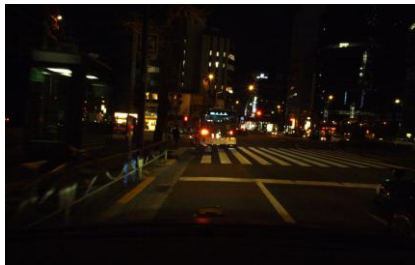
test_486

test_103

test_575

タイプ1

元画像

結果
貼り合
わせ後左結果
貼り合
わせ前右結果
貼り合
わせ前エラー
箇所

test_205

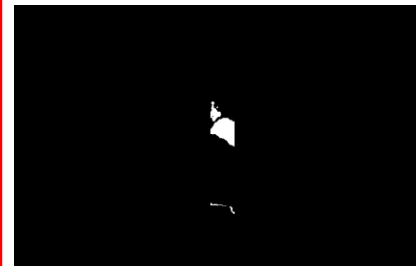
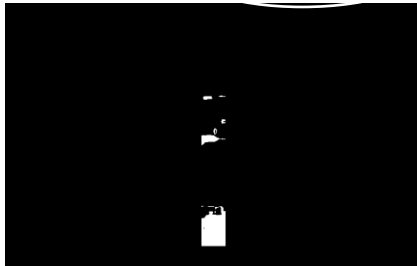
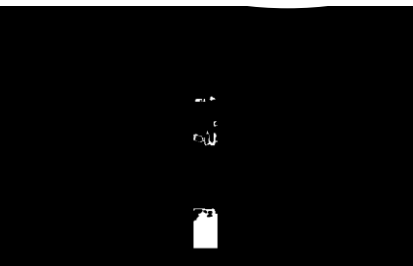
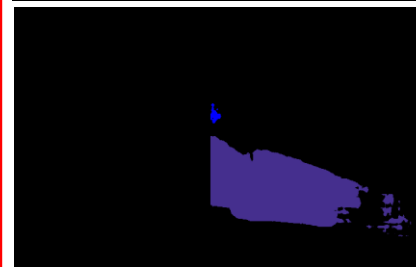
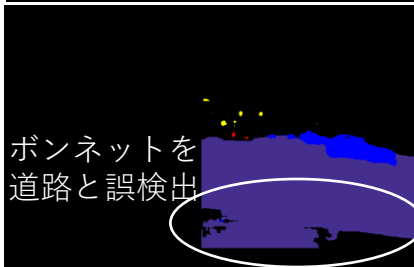
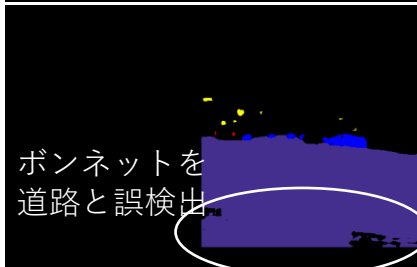
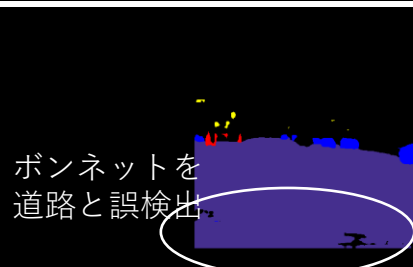
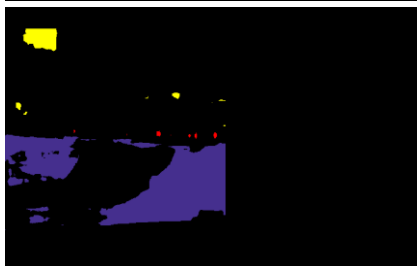
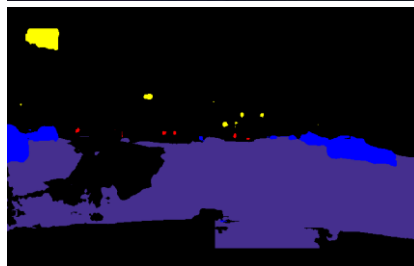
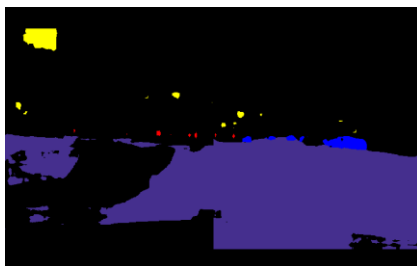
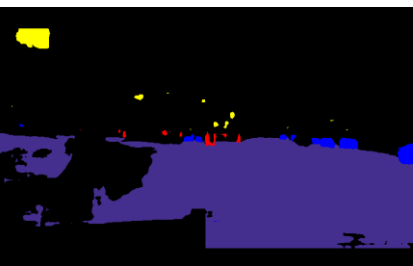
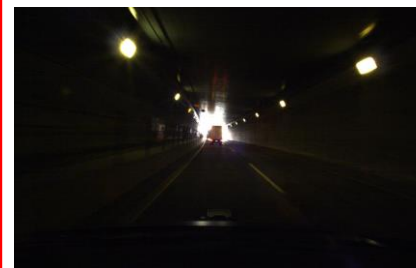
test_413

test_441

test_362

タイプ2

元画像

結果
貼り合
わせ後左結果
貼り合
わせ前右結果
貼り合
わせ前エラー
箇所ボンネットを
道路と誤検出ボンネットを
道路と誤検出ボンネットを
道路と誤検出トンネル出口
を道路と誤検出ほぼ
同一条件

test_064

test_306

元画像

結果
貼り合
わせ後

左結果
貼り合
わせ前

右結果
貼り合
わせ前

エラー
箇所

