

セマンティックセグメンテーションにおける Skip Connect手法の比較

岡山大学 大学院環境生命科学研究科

景山 晶

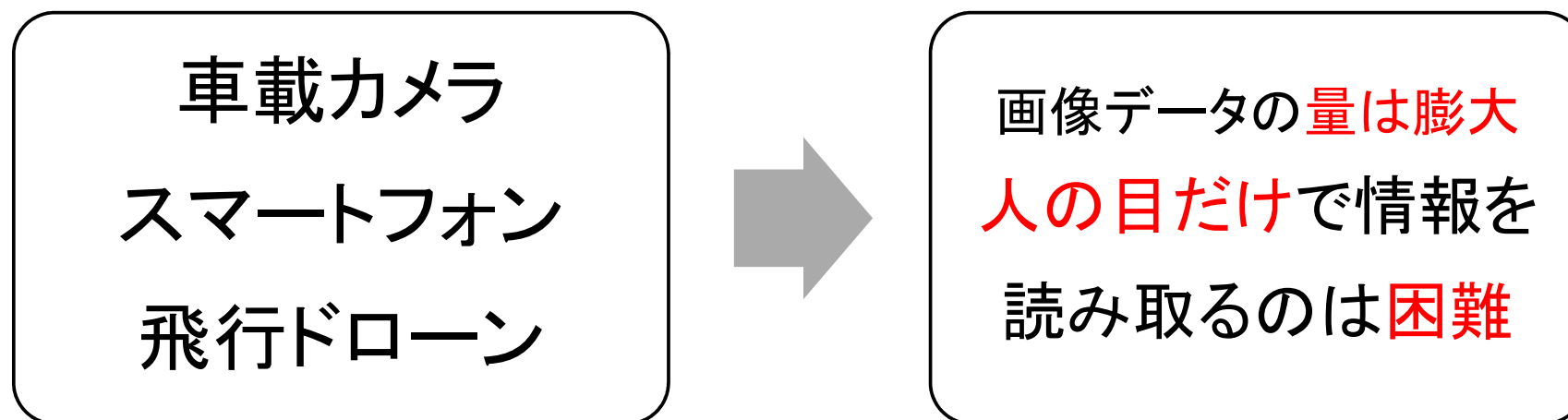
坂本 亘

目次

- 1.背景
- 2.事前知識共有
- 3.先行研究
- 4.提案手法
- 5.実験
- 6.結論・課題

背景（画像データの氾濫）

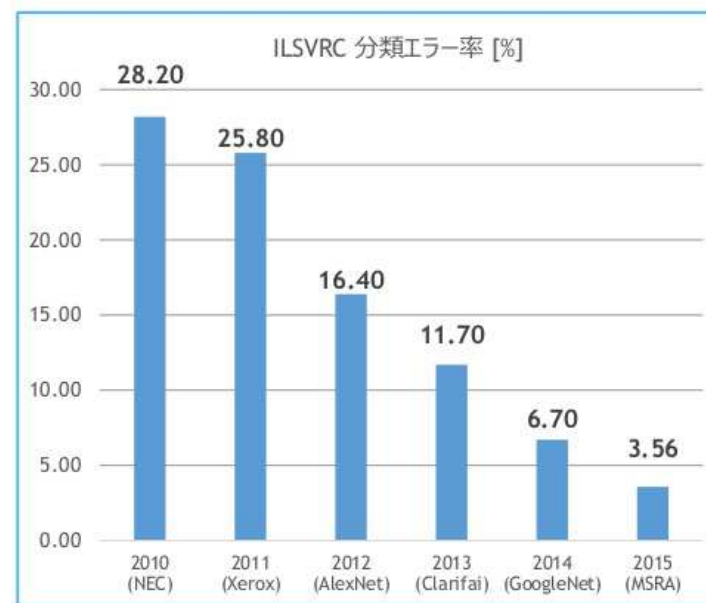
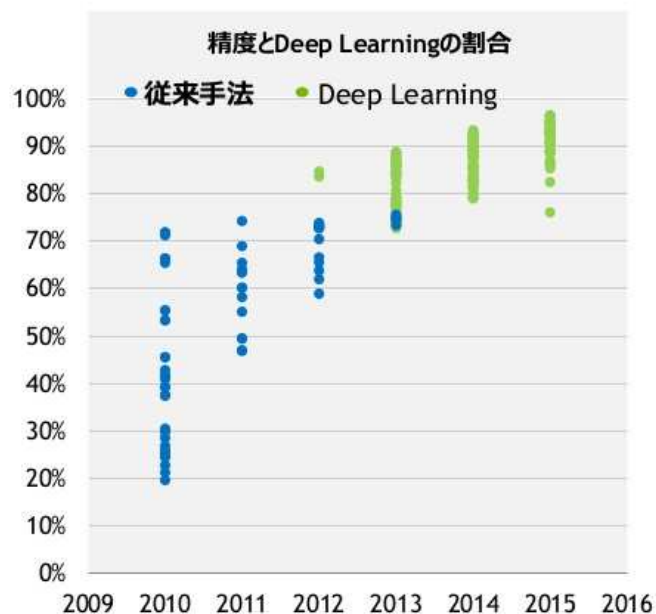
現代では多くの画像データが利用できるように。



画像データに対して統計学や機械学習のアプローチを採用したい。

背景(深層学習モデルの台頭)

ImageNet の結果の推移



井崎武士 広がるディープラーニングの活用, NVIDIA JAPAN
<<https://www.slideshare.net/NVIDIAJapan/ss-57361969>>

セマンティック・セグメンテーション

直訳: **意味+分割**

画像の**ピクセル1つ1つ**に何が写っているかを推定する手法.

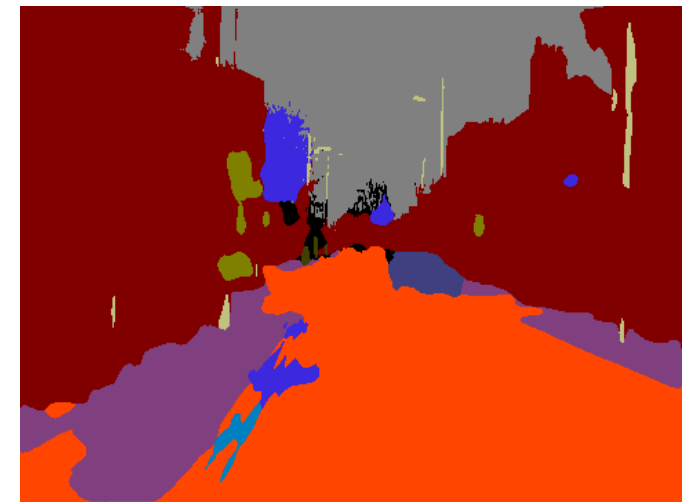
自動運転や**医用画像診断**などへの応用が期待される.



入力



モデルによる推定



出力

SegNet and Bayesian SegNet Tutorial
<<https://github.com/alexgkendall/SegNet-Tutorial>>

目次

- 1.背景
- 2.事前知識共有
- 3.先行研究
- 4.提案手法
- 5.実験
- 6.結論・課題

固定基底関数モデル

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \sigma \left(\sum_{j=1}^D w_j \varphi_j(\mathbf{x}) + w_0 \right) \quad (2.1)$$

D 次元の説明変数: \mathbf{x}

目的変数 \mathbf{y}

パラメータ: \mathbf{w}

任意の基底関数: φ

活性化関数: σ (恒等写像:回帰問題、シグモイド関数:2クラス分類問題)

参考文献:Bishop 2006

ニューラルネットワーク

Neural Network(以下NN)

基底関数 φ_j をパラメトリックな線形和で表したものの
2層ニューラルネットワークの場合以下のようにかける。

$$y_l(\mathbf{x}, \mathbf{w}) = \sigma \left(\sum_{j=1}^M w_{lj}^{(2)} h \left(\sum_{i=1}^D w_{ij}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{l0}^{(2)} \right) \quad (2.2)$$

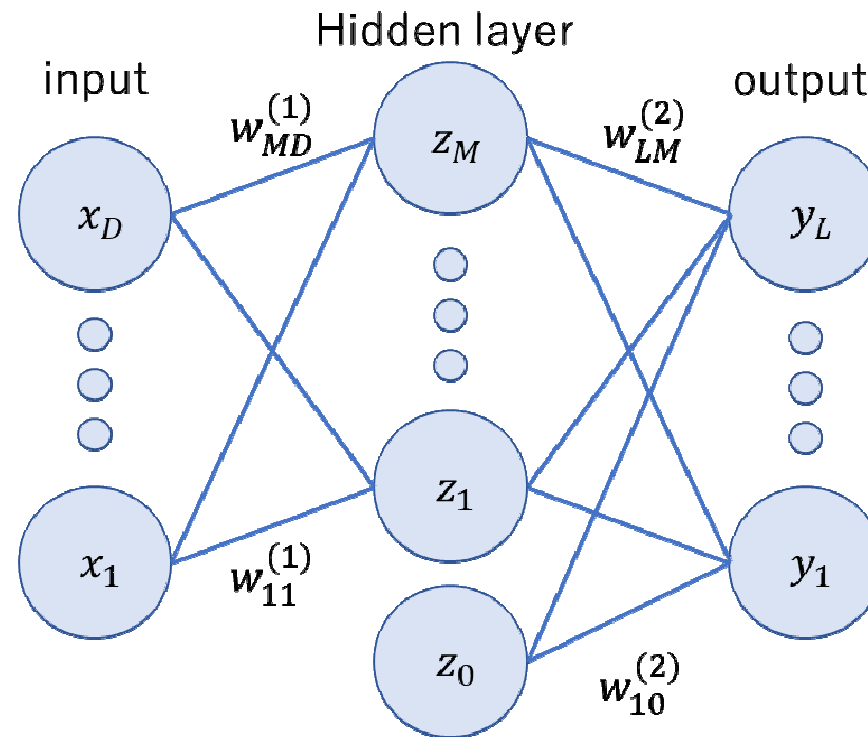
分類クラス: $l(1, \dots, L)$

活性化関数: h, σ

2層目の説明変数の数: M

ニューラルネットワーク図解

2layer Dense Neural Network



ニューラルネットワークによる多クラス分類

問題設定

- 訓練集合の説明変数 $x_n (n = 1, 2, \dots, N)$
- L 個のクラスに分類する

L 次元のターゲットベクトル t_n (既知の目的変数)

- 分類したいクラスの成分1, そうでない成分0

例: l 番目が真のクラスの場合

$$\mathbf{t}_n = (0, \dots, \underset{l\text{番目}}{1}, \dots, 0)^T \quad (2.3)$$

ベイズの定理とクラス分類

あるクラス $i(1,2, \dots, L)$ について

$$a_{ni} = \ln p(\mathbf{x}_n | t_{ni} = 1) p(t_{ni} = 1) \quad (2.4)$$

とするとベイズの定理より,

$$y_{ni}(\mathbf{x}_n, \mathbf{w}_i) = p(t_{ni} = 1 | \mathbf{x}_n)$$

データ点 \mathbf{x}_n が与えられたうえで
クラスが i である確率

$$= \frac{p(\mathbf{x}_n | t_{ni} = 1) p(t_{ni} = 1)}{\sum_{l=1}^L p(\mathbf{x}_n | t_{nl} = 1) p(t_{nl} = 1)} \quad (2.5)$$

$$= \frac{\exp(a_i(\mathbf{x}_n, \mathbf{w}_i))}{\sum_{l=1}^L \exp(a_l(\mathbf{x}_n, \mathbf{w}_i))}$$

a_i の計算に必要な部分のパラメータ: \mathbf{w}_i

ソフトマックス関数

クロスエントロピー誤差関数

データ点が独立同分布に得られたと仮定.

この時尤度関数は

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^N \sum_{l=1}^L y_{nl}^{t_{nl}} \quad (2.6)$$

誤差関数(負の対数尤度)は

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{l=1}^L t_{nl} \ln y_{nl}(\mathbf{x}_n, \mathbf{w}) \quad (2.7)$$

クロスエントロピー誤差関数と呼ぶ.

クロスエントロピー誤差関数の勾配

$a_n = \mathbf{w}^T \mathbf{x}_n$ とすると, 誤差関数の勾配は

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = (\mathbf{y}_n - \mathbf{t}_n)^T \mathbf{x}_n \quad (2.8)$$

とかける.

参考webサイト: 多クラス交差エントロピー誤差関数とソフトマックス関数, その美しい微分
<<https://qiita.com/klis/items/4ad3032d02ff815e09e6>>

パラメータ最適化

誤差関数は非線形→解析的に解くのは難しい

- 数値的反復手段を用いる

勾配降下法

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \quad (2.9)$$

パラメータの更新回数: τ

学習率: η

- 設定する必要があるハイパーパラメータ.

確率的勾配降下法

訓練集合のすべてを用いて推論→計算資源のコストがかかる

- メモリ等

逐次的にデータ点1つごとに勾配を更新する

独立同分布の仮定の下では誤差関数は

$$E(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w}) \quad (2.10)$$

と和の形でかけるので、

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)}) \quad (2.11)$$

とデータ点毎に更新する。

ミニバッチ最適化

訓練集合全体: $\{x_1, \dots, x_N\}$

サイズ m の部分集合 $\{x_1, \dots, x_m\}$

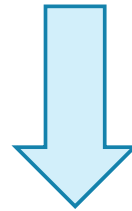
部分集合ごとにパラメータを更新

- 部分集合の選び方のランダム性で停留点から脱出
- メモリの消費を抑えられる

畳み込みニューラルネットワーク

画像データ特有の問題

- 画像データでは画素の1つ1つが全て説明変数
- すべての説明変数の線形和をとると**パラメータの数**が膨大に



パラメータの数を削減する工夫が必要

参考文献:原田達也(2017), 画像認識, 講談社.

特徴マップ

画像データは2次元の配列状のデータ

- その空間的特性を活かしたい

各ユニットを2次元の配列状に並べ, 位置の情報を保持する
そしてその平面の数をチャンネルと呼ぶ.

例

デジタルのフルカラー写真

- RGBの3チャンネル



SegNet and Bayesian SegNet Tutorial

<https://github.com/alexkendall/SegNet-Tutorial> より引用し加工したもの

畳み込み層

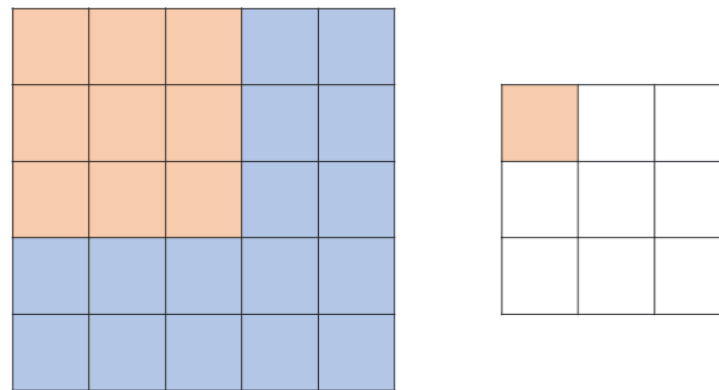
画像データに対して以下の2つの仮定をする

- 1、画素同士の関係は**お互いの距離が近いものほど強い**
- 2、局所的な特徴は**画像の位置によらず共通**とする

1の仮定→注目画素の周辺のみで線形和を取る

2の仮定→注目画素の場所が異なっても同じパラメータを使う

上記の仮定を用いる層を**畳み込み層**と呼ぶ。



畳み込み層のパラメータ

畳み込みを行うパラメータは3次元のテンソルの集まり

- 縦横の2次元
- 入力ユニットのチャンネルの数の1次元
- このテンソル1つから次の層の1チャンネルが計算される

任意の数のテンソルを用意

その数だけ次の層のチャンネルが計算される

- モデルの表現力は向上する
- パラメータの数が増大する

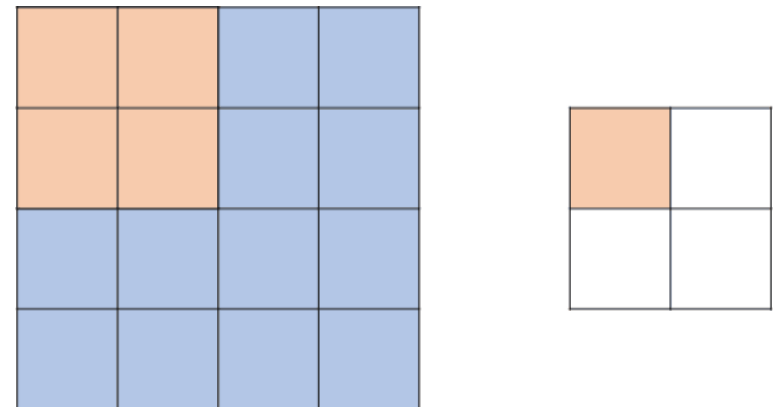
プーリング層

特徴マップには局所的な歪みや平行移動の影響がある

細かな歪みに対処するために局所的領域の統計量を計算する

畳み込みと異なり、領域は重ならないように取る

統計量には平均値や最大値が用いられる



畳み込み層の順伝播 1

変数等一覧

ある中間層 j を中心に考える

特徴マップ: $M^{(j)}$ = 幅, $N^{(j)}$ = 高さ, $K^{(j)}$ = チャンネルの数

- $z_{m,n,k}^{(j)}$: j 層の (m, n) の位置にあるチャンネル k の出力

パラメータテンソル: P^{j+1} = 幅, Q^{j+1} = 高さ, K^j = チャンネルの数

- $K^{(j+1)}$ 個用意する
- $w_{p,q,k,k'}^{(j+1)}$: k' 番目のテンソルのチャンネル k の位置 (p, q) の重み

畳み込み層の順伝播 2

$j + 1$ 層の入力は以下のように計算できる.

$$a_{m,n,k'}^{(j+1)} = \sum_{p,q,k} w_{p,q,k,k'}^{(j+1)} z_{m+p,n+q,k}^{(j)} \quad (2.12)$$

活性化を行い, $j + 1$ 層からの出力は,

$$z_{m,n,k'}^{(j+1)} = h(a_{m,n,k'}^{(j+1)}) \quad (2.13)$$

とかける.

畳み込み層の逆伝播1

誤差関数のパラメータ w に関する勾配を考える

$$\frac{\partial E}{\partial w_{p,q,k,k'}^{(j)}} = \sum_{m,n} \frac{\partial E}{\partial a_{m,n,k'}^{(j)}} \frac{\partial a_{m,n,k'}^{(j)}}{\partial w_{p,q,k,k'}^{(j)}} \quad (2.14)$$

右辺の積の右側は

$$\frac{\partial a_{m,n,k'}^{(j)}}{\partial w_{p,q,k,k'}^{(j)}} = z_{m+p,n+q,k}^{(j-1)} \quad (2.15)$$

畳み込み層の逆伝播2

積の左側を

$$\frac{\partial E}{\partial a_{m,n,k'}^{(j)}} = \delta_{m,n,k'}^{(j)} \quad (2.16)$$

と定義し置くことで誤差関数の勾配は

$$\frac{\partial E}{\partial w_{p,q,k,k'}^{(j)}} = \sum_{m,n} \delta_{m,n,k'}^{(j)} z_{m+p,n+q,k}^{(j-1)} \quad (2.17)$$

とかける。

畳み込み層の逆伝播3

$\delta_{m,n,k}^{(j)}$ に微分の連鎖律を用いて,

$$\delta_{m,n,k}^{(j)} = \frac{\partial E}{\partial a_{m,n,k}^{(j)}} = \sum_{p,q,k'} \frac{\partial E}{\partial a_{m-p,n-q,k'}^{(j+1)}} \frac{\partial a_{m-p,n-q,k'}^{(j+1)}}{\partial a_{m,n,k}^{(j)}} \quad (2.18)$$

左側は $\delta_{m-p,n-q,k'}^{(j+1)}$

右側は

$$\frac{\partial a_{m-p,n-q,k'}^{(j+1)}}{\partial a_{m,n,k}^{(j)}} = w_{p,q,k,k'}^{(j+1)} h' \left(a_{m,n,k}^{(j)} \right) \quad (2.19)$$

よって

畳み込み層の逆伝播4

$$\delta_{m,n,k}^{(j)} = h' \left(a_{m,n,k}^{(j)} \right) \sum_{p,q,k'} \delta_{m-p,n-q,k'}^{(j+1)} w_{p,q,k,k'}^{(j+1)} \quad (2.20)$$

出力側の誤差 $\delta^{(j+1)}$ が得られれば

入力側の誤差 $\delta^{(j)}$ が計算できる.

クロスエントロピー誤差関数の場合の勾配は

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = (\mathbf{y}_n - \mathbf{t}_n)^T \mathbf{x}_n \quad (2.8)$$

とかけていたので、出力側の誤差 $\delta^{(j+1)}$ はデータがあれば計算可能

※式2.8のnはデータ点の添字

プーリング層の順伝播と逆伝播

最大値プーリングの場合

【順伝播】

$R_{m,n}$: 位置 m, n に出力するプーリングを適用する小領域

$$z_{m,n,k}^{(j+1)} = \max_{(u,v) \in R_{m,n}} z_{u,v,k}^{(j)} \quad (2.21)$$

【逆伝播】

最大値プーリングで選ばれた位置に誤差を返す.

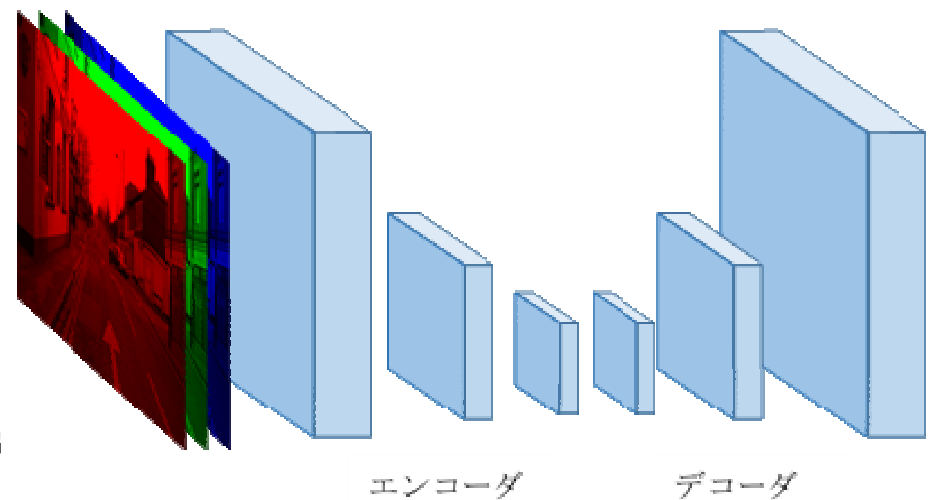
エンコーダ-デコーダ

畳み込みニューラルネットワークはプーリングする

- 解像度が悪くなる
- 何らかの方法で拡大して出力する必要がある

入力をエンコードし特徴抽出

拡大と畳み込みを繰り返し元の解像度までデコード



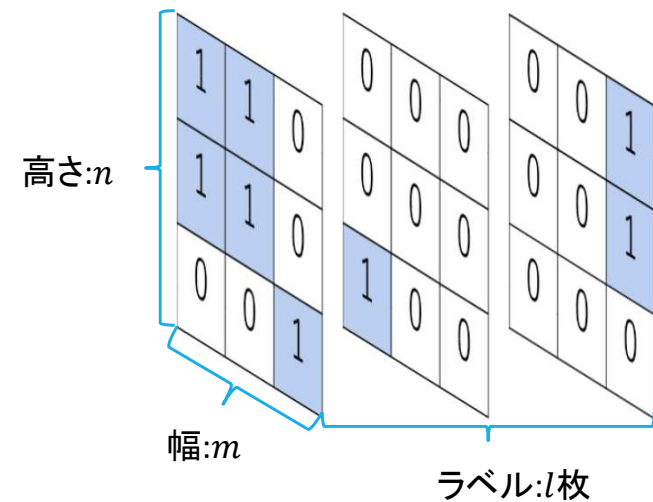
参考文献:太田満久/須藤広大/黒澤匠雅/小田大輔著
(2018),現場で使える!TensorFlow開発入門 kerasによる深層
学習モデル構築手法,翔泳社.

セマンティックセグメンテーション

画素の1つ1つに対して分類問題を解く

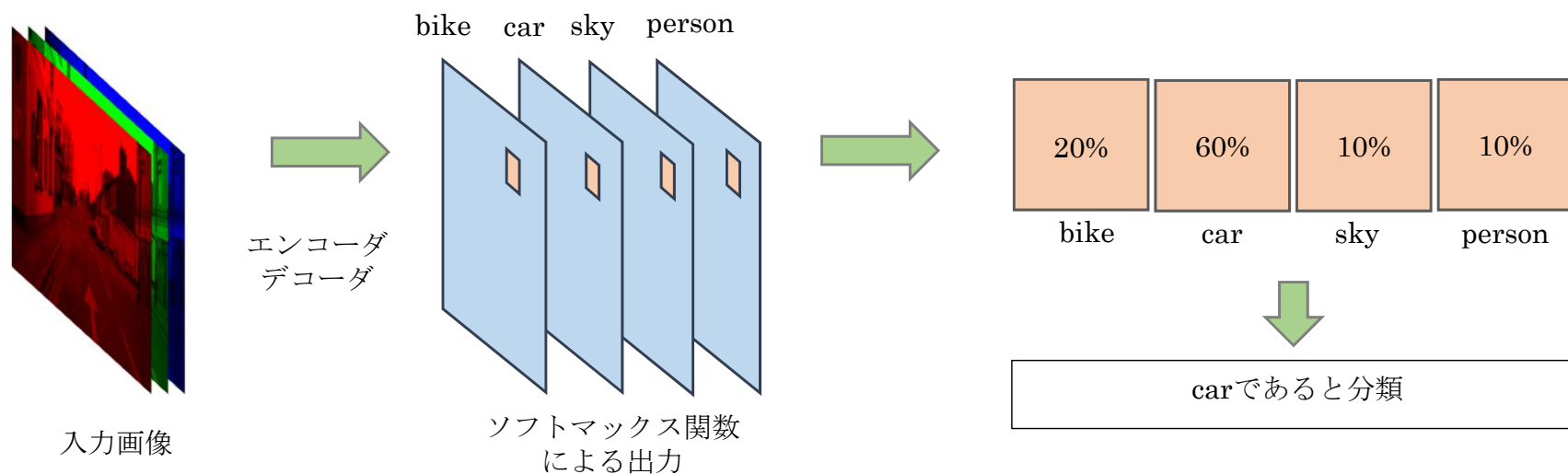
説明変数: RGBの3チャンネル画像データ

目的変数: ラベルが真なら1, 偽なら0の特徴マップを
ラベルの種類だけ集めたテンソル



SSにおける分類問題の例図

ピクセルごとにラベル方向で多クラス分類問題を解く



性能評価指標(Accuracy)

真の予測したいラベルの数:Ground Truth(GT)

モデルの予測が的中した部分の数:True Positive(TP)

真のクラスが*l*の領域に関する**正識別率(Accuracy)**

$$Accuracy_l = \frac{TP_l}{GT_l} \quad (2.22)$$

参考文献「画像認識をやってみました」の次にすること

<https://www.sigfoss.com/developer_blog/detail?actual_object_id=147>

性能評価指標(Global Accuracy)

データセット全体のGTの数に対するTPの割合

- Global Accuracy(GA)

$$GA = \frac{\sum_{l=1}^L TP_l}{\sum_{l=1}^L GT_l} \quad (2.23)$$

- 特定のクラスの分類性能が悪くても
ピクセルの数が多いクラスを予測すれば結果がよくなりやすい

性能評価指標(Class Accuracy)

Accuracyのクラス平均をとったClass Accuracy(CA)

$$CA = \frac{1}{L} \sum_{l=1}^L \frac{TP_l}{GT_l} \quad (2.24)$$

- ピクセルの総数が少ないクラスも正確に分類できないと悪化しやすい

性能評価指標(mIoU)

セマンティックセグメンテーションは輪郭を予測したい

- Accuracyは予測して誤ったことを考慮に入れていない

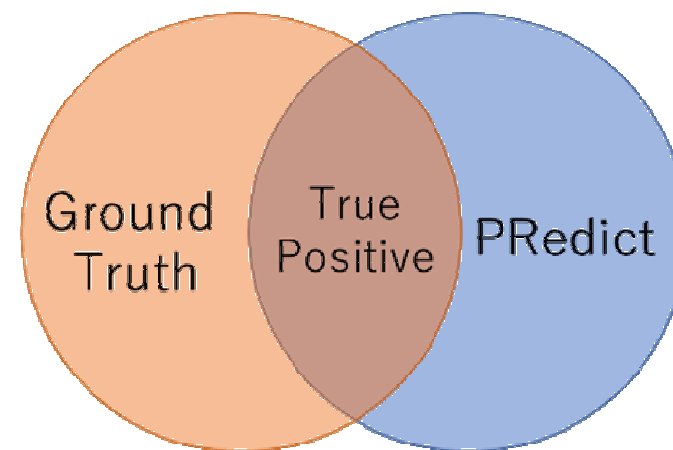
IoU(Intersection over Union)

- 予測した領域の数: PRedict(PR)を考慮する

$$IoU_l = \frac{TP_l}{GT_l + PR_l - TP_l} \quad (2.25)$$

- クラス平均を取る

$$mIoU = \frac{1}{L} \sum_{l=1}^L IoU_l \quad (2.26)$$



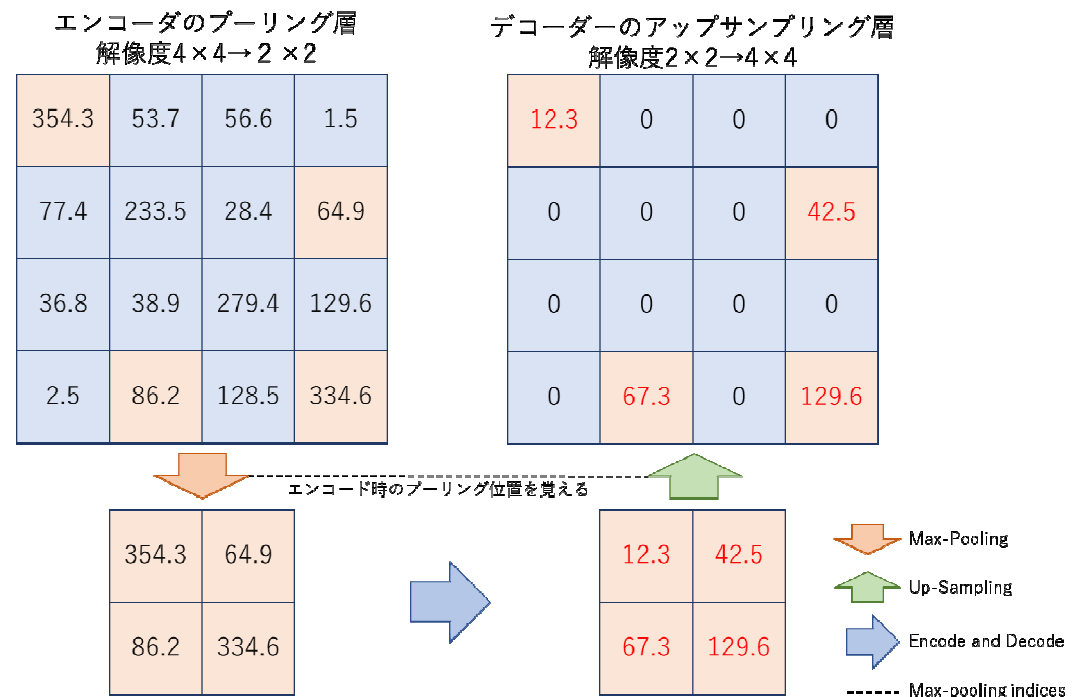
目次

- 1.背景
- 2.事前知識共有
- 3.先行研究
- 4.提案手法
- 5.実験
- 6.結論・課題

SegNet (Badrinarayanan et al. 2015)

Max-pooling indices

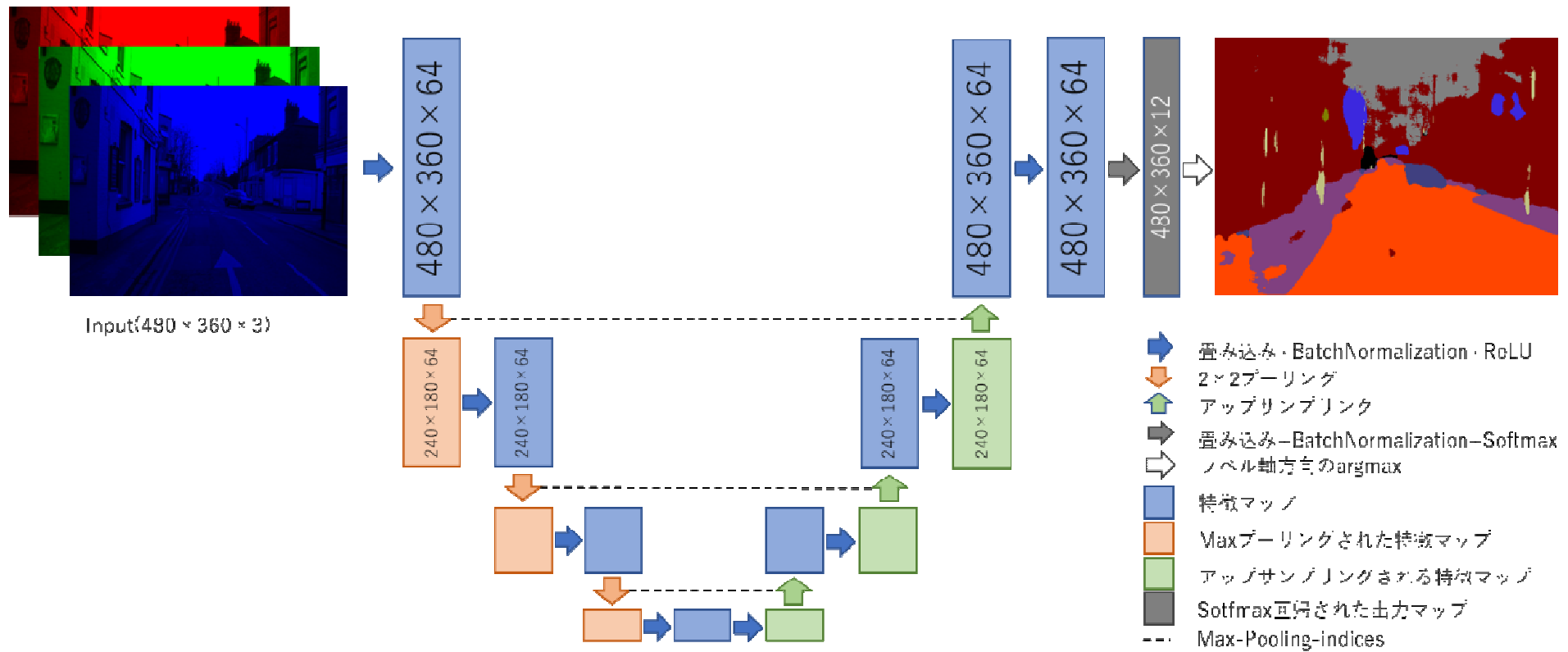
- 最大値プーリングの際にプーリング位置を記録
- デコードの際にその位置へアップサンプリングを行う
- プーリング位置の情報を保持しながらアップサンプリングできる
- 他の位置には0を入れる



SegNetBasic

モデルのパラメータや層の数を抑えた簡易版のモデル

- 図は私が再現したモデル



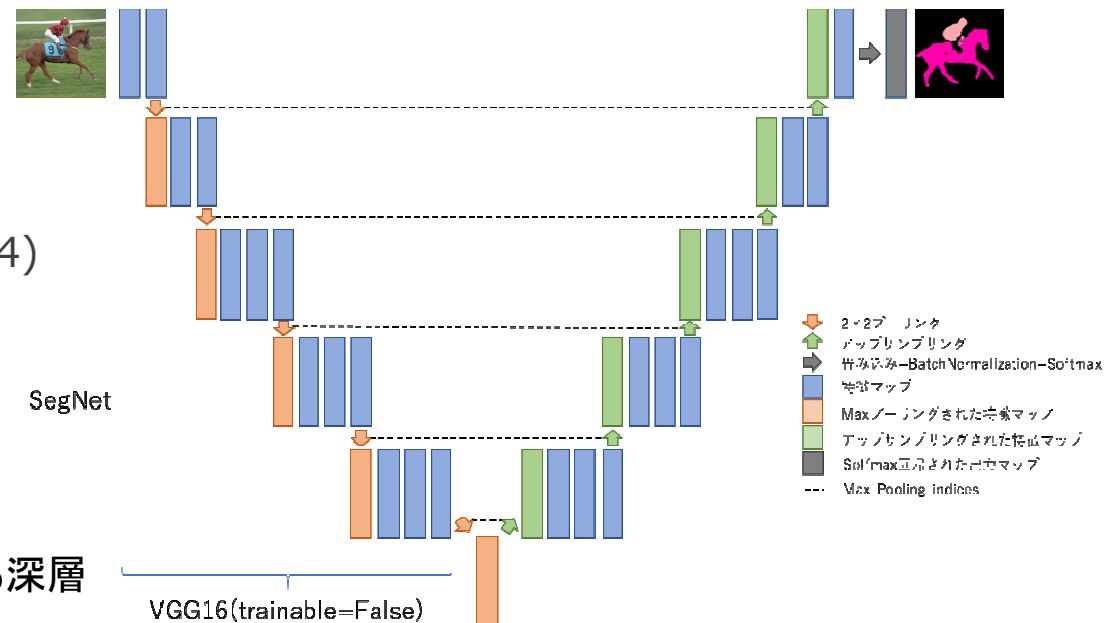
SegNetと転移学習

何らかのデータセットで学習済みのモデルに着目

- 膨大なデータで学習済みであれば特徴抽出機として使える
- 分類を行う層を問題に合わせて増設
- 膨大な計算資源やデータのコストを削減

左半分が転移個所 →

- VGG16(ImageNetで学習)
 - Oxford大学のチームのモデル(2014)
- TrainableをFalseにする
- 右半分は増設する



参考文献:太田満久/須藤広大/黒澤匠雅/小田大輔著
(2018),現場で使える!TensorFlow開発入門 kerasによる深層
学習モデル構築手法,翔泳社.

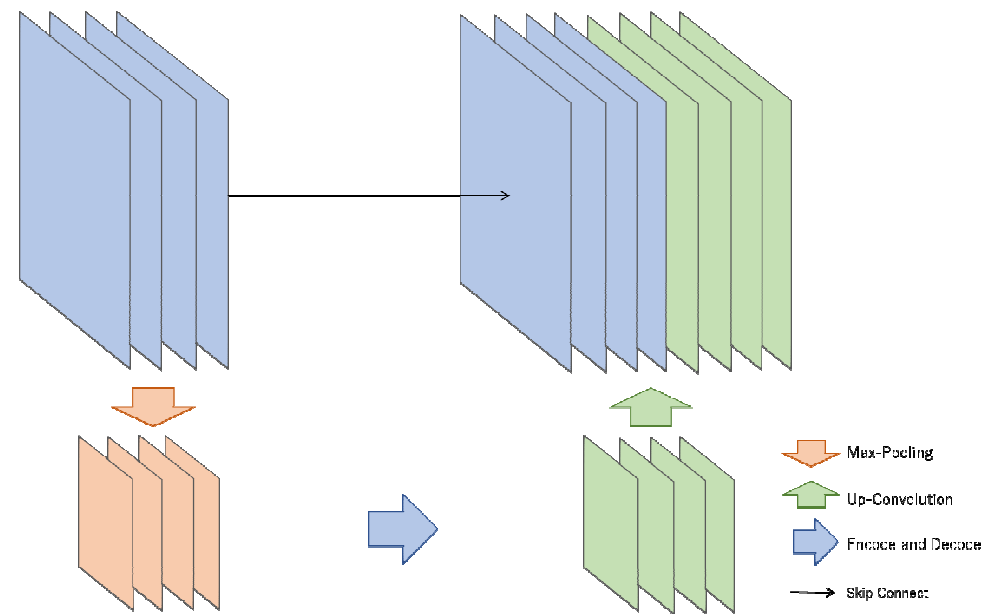
UNet(Ronneberger et al. 2015)

Contracting pathが提案された

- 低次元の特徴を高次元の特徴へスキップして結合する

UNetではConcatenateと呼ばれる方法で結合

- 結合対象の2つのテンソルを1つの大きなテンソルとして結合
- チャンネルの数が2倍になる
- パラメータが増加する



Concatenateのパラメータ増加

畳み込みカーネルの一片の大きさ: k

Contracting pathが無い場合の元々のチャンネルの数: c

次の層(c 枚)へのパラメータの総数

$$k \times k \times c \times c = k^2 c^2 \quad (3.1)$$

Concatenate後のチャンネル数: $2c$

次の層(c 枚)へのパラメータの総数

$$k \times k \times 2c \times c = 2k^2 c^2 \quad (3.2)$$

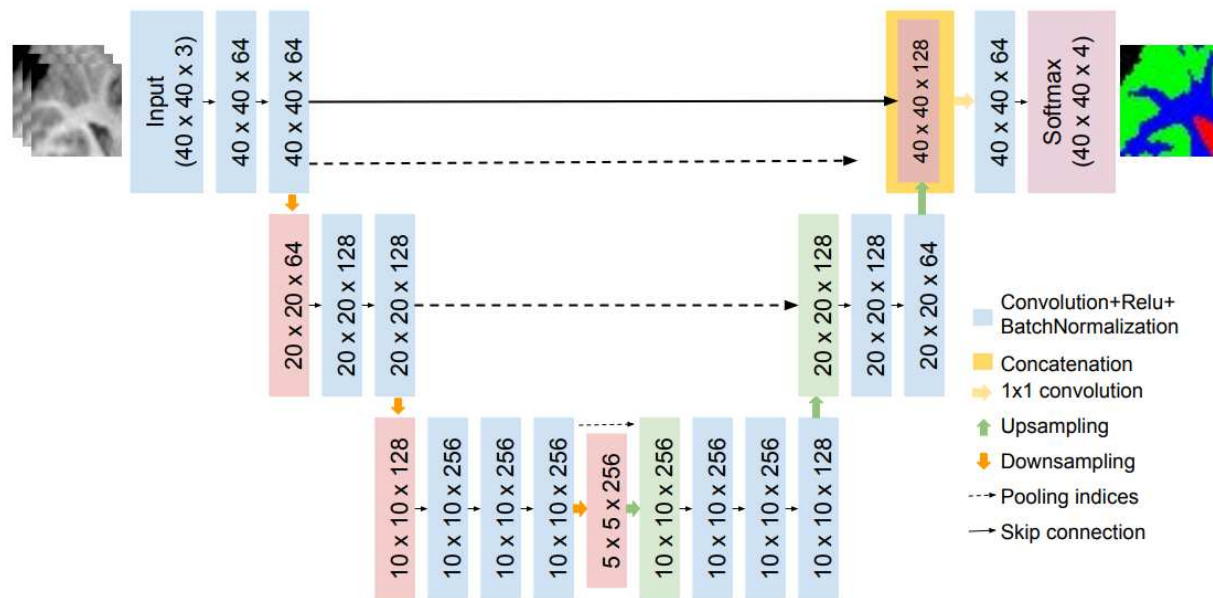
Concatenateするとパラメータが2倍必要

- 過学習や推論時間の増加が懸念される

USegNet(Kumar et al. 2018)

SegNetとUNetのハイブリッドモデル

- Max-pooling indices
- Contracting path(USegNetではSkip Connectionと呼称)
- これらは同時に採用することができる



USegNetの図

Pulkit Kumar, et al(2018). U-SEGNET: FULLY CONVOLUTIONAL NEURAL NETWORK BASED AUTOMATED BRAIN TISSUE SEGMENTATION TOOL.arXiv:1806.04429.

USegNetでのパラメータ削減の工夫

Skip connectionにはConcatenateを使用

Concatenate後, 1×1 畳み込みを行う

- チャンネル方向の線形和

Concatenate前のチャンネル数を c 枚とするとパラメータの数は

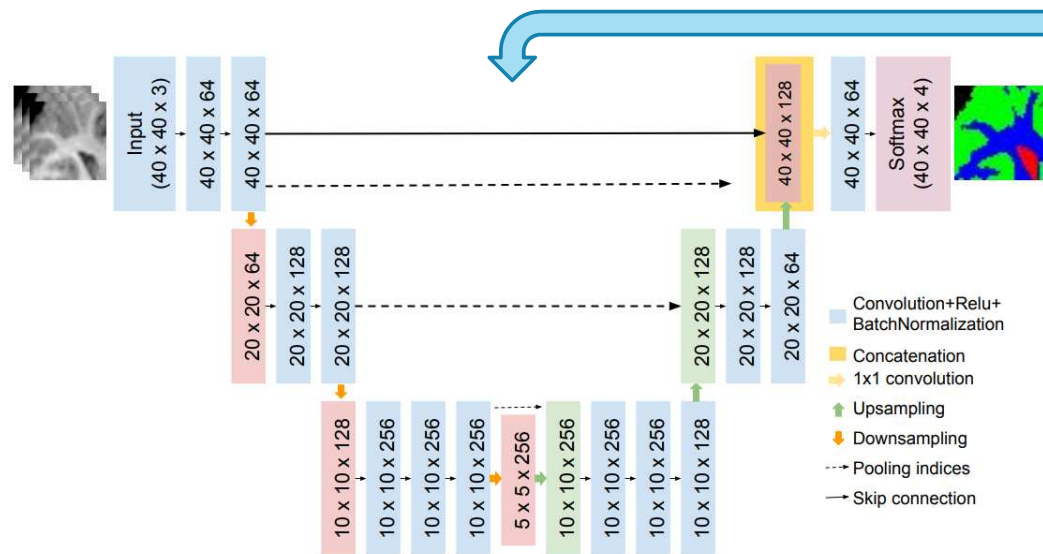
$$1 \times 2c \times 1 \times c = 2 \times 1^2 c^2 \quad (3.3)$$

となり, パラメータの増加を抑えることができる.

Skip connectionの接続箇所

Skip connection

- チャンネルの解像度が同じであれば接続できる
- 何か所接続するかが問題
- USegNetの論文内では**最大解像度のみ**に接続した場合が最良
- パラメータの数が増大するからではないかと記載



解像度が
最大の箇所のみ
接続

ResNet(He et al.2015)

一般的な畳み込みニューラルネットワークの研究

- エンコーダ-デコーダではない

畳み込みニューラルネットワーク

- 層を深くしすぎると性能悪化

層を増やして悪化する→恒等写像より悪い結果

- 恒等写像を表現しやすくしたい

Shortcut Connectionが提案された

Shortcut connection

真の推定したいモデル構造を $H(\mathbf{x})$ とする

- $H(\mathbf{x})$ はベクトルであり \mathbf{x} と同じ次元を持つ

$H(\mathbf{x})$ と \mathbf{x} の差を

$$F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x} \quad (3.4)$$

とし, $F(\mathbf{x})$ がアルゴリズムで近似できるのであれば

$$H(\mathbf{x}) = F(\mathbf{x}) + \mathbf{x} \quad (3.5)$$

を近似することができる.

式(3.5)は $F(\mathbf{x})$ のパラメータを全て0にすることで恒等写像になる

- 直接 $H(\mathbf{x})$ を推定するよりも恒等写像の表現が容易に

Skip と Shortcut

Skip ConnectionとShortcut connectionを比較

D次元の関数 $F(x)$ とベクトル x に当てはめる

Skip connection(USegNet)	Shortcut connection(ResNet)
$(F_1(x), \dots, F_D(x), x_1, \dots, x_D)^T$	$F(x) + x$

- Skip connectionはベクトルの成分を増やす操作
- 恒等写像の表現はSkip Connectionではできていない。

既存手法まとめ

モデル	プーリング情報	低次元特徴の利用	恒等写像の表現	パラメータの増加
SegNet	○	×	×	Zero
UNet	×	○	×	$k^2 c^2$
USegNet	○	○	×	c^2
ResNet (非セマセグ)	×	○	○	Zero



USegNetのSkip Connectionに
Shortcut Connectionの採用を提案する

目次

- 1.背景
- 2.事前知識共有
- 3.先行研究
- 4.提案手法
- 5.実験
- 6.結論・課題

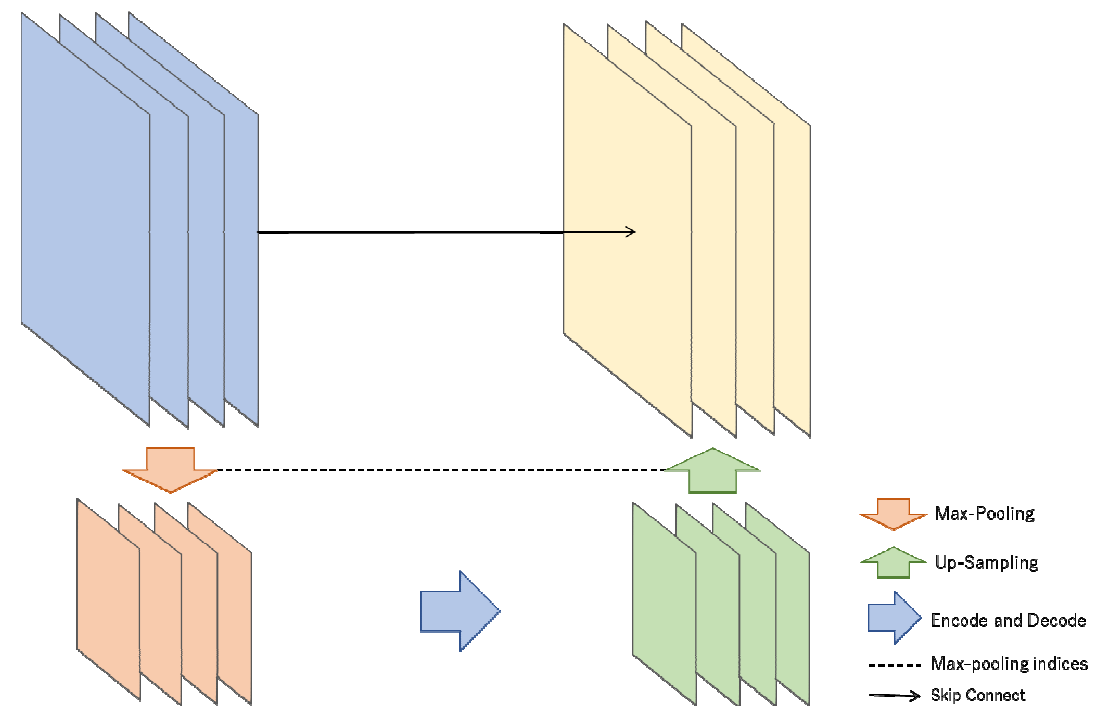
USegNetAdd(提案手法)

Skip Connectionの結合方法に和(Add)を用いる

- パラメータの増加が0個
- Shortcut Connectionによる恒等写像の表現

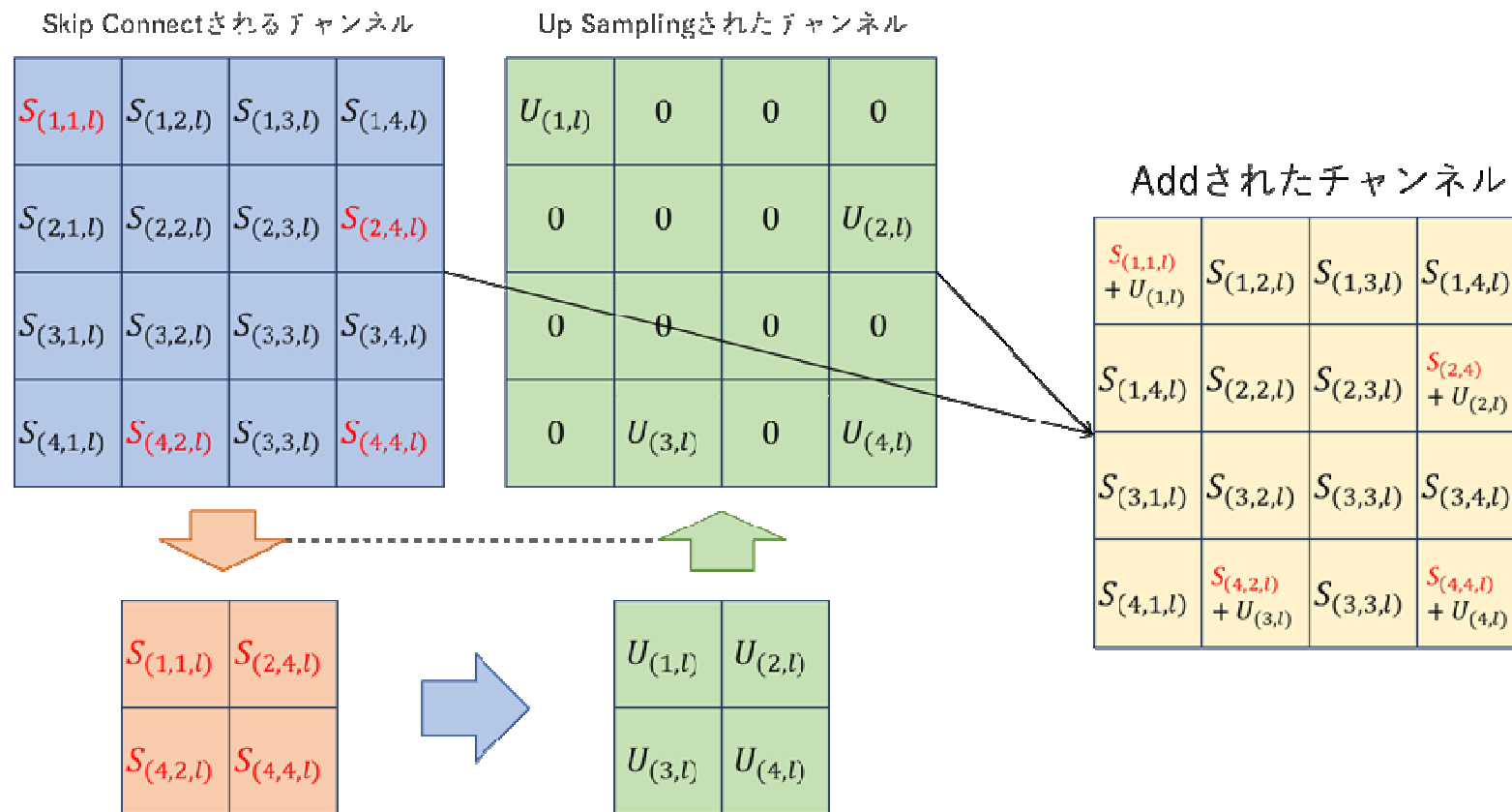
さらなるメリット？

- チャンネル位置情報の保持
- 境界情報の保持



Addの処理

Max-pooling indicesとAdd結合の処理



式で見るAdd結合

Shortcut Connectionされる3次元の特徴テンソル: S

- 位置: m, n ($m=1, \dots, M, n=1, \dots, N,$)
- チャンネル: l ($l=1, \dots, L$)
- 特定の値: $S_{n,m,l}$

アップサンプリングされてきた特徴テンソル: U

- 位置やチャンネルは S と同様に m, n, l で表記

S を変換し U を求める関数を $F(S)=U$ とする

Shortcut connectionでの潜在的な関数 H は

$$H(S) = F(S) + S = U + S \quad (4.1)$$

チャンネル位置情報の保持

AddはConcatenateより多くの情報を保持できる

Addの場合の $H(S)$ の mn 成分

$$H_{mnl}(S) = U_{mnl} + S_{mnl} \quad (4.2)$$

- S と U のチャンネルの位置 l が等しい
- チャンネル位置情報を保持

Concatenateの場合の $H(S)$ の mn 成分は

$$H(S)_{m,n} = (U_{mn1}, \dots, U_{mnl}, S_{mn1}, \dots, S_{mnl})^T \quad (4.3)$$

- チャンネル位置 l は U と S の間で対応付けが消失

境界情報の保持

被写体の**クラス**の境界は**色の変化が大きい**と仮定

- Max-pooling indices はその境界情報を捉える
- デコーダではその境界位置へアップサンプリング

Skip Connectionの時, その位置へAddする

- **特定のチャンネル, 位置が最大であった**という情報が維持されやすい?

提案手法の位置づけ

モデル	プーリング情報	低次元特徴の利用	恒等写像の表現	パラメータの増加	チャンネル位置の保持	境界情報の保持
SegNet	○	×	×	Zero	×	○
UNet	×	○	×	$k^2 c^2$	×	×
USegNet	○	○	×	c^2	×	○
ResNet (非セマセグ)	×	○	○	Zero	○	×
USegAdd	○	○	○	Zero	○	◎

実験で確かめる

目次

- 1.背景
- 2.事前知識共有
- 3.先行研究
- 4.提案手法
- 5.実験
- 6.結論・課題

実験方法

2種類のデータセットに対して実験を行った

- CamVidデータセット(車載カメラ)
 - SegNetBasicで使用された
- PASCAL VOC2012(一般的オブジェクト:犬や飛行機など)

環境

- OS:Windows10
- CPU:intel i7 8700
- GPU:GTX1060(6GB)
- 言語:Python3.6
- 深層学習ライブラリ:Tensorflow-gpu1.10.0, Keras2.2.4

実験方法

データセット全体をtrain, validation, testに分ける.

- trainデータを用いてモデルを学習させる.

最適化法:ミニバッチサイズ4の勾配降下法

- GPUメモリの都合上ミニバッチサイズを大きくできませんでした
- 学習率0.1, 慣性項の係数を0.9とする.(SegNetBasicを参考した)

trainデータ全体の非復元抽出を行う.

- これを100epochs(回)繰り返す.

各epochs後validationデータのGAを計算

- 100回の中で最もGAが高かったepochsのモデルを選択.

選択したモデルを用いてtestデータでのGA, CA, mIoUを評価.

実験1: CamVidデータセット

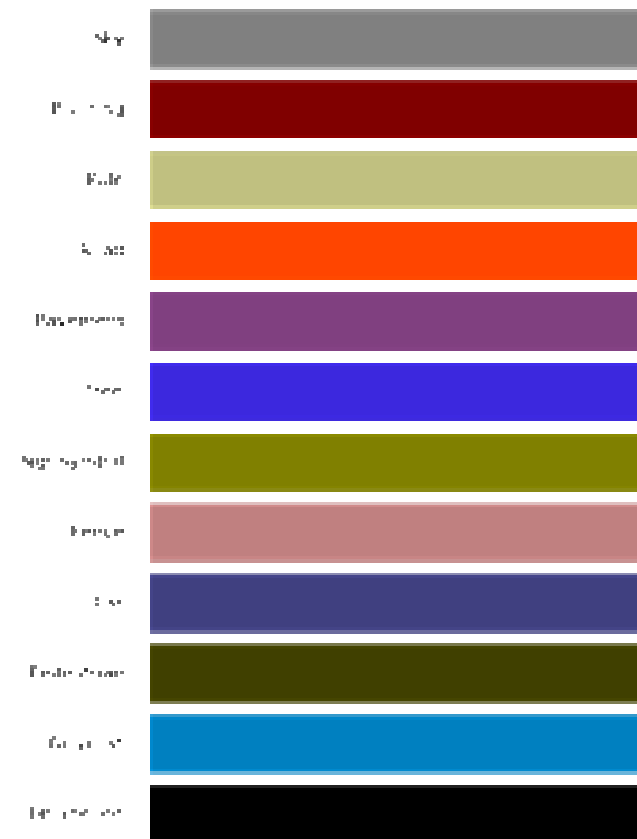
ケンブリッジの車載カメラ視点ラベル付きビデオデータベース

SegNetの論文内で使用されている

- 論文内ではデータセットの一部を使用
- ラベルについていくつかを統合
- 11のクラスとUnlabeledがある
- 解像度は480×360の横長

サンプルサイズ

- train:367枚
- validaton:101枚
- test:233枚

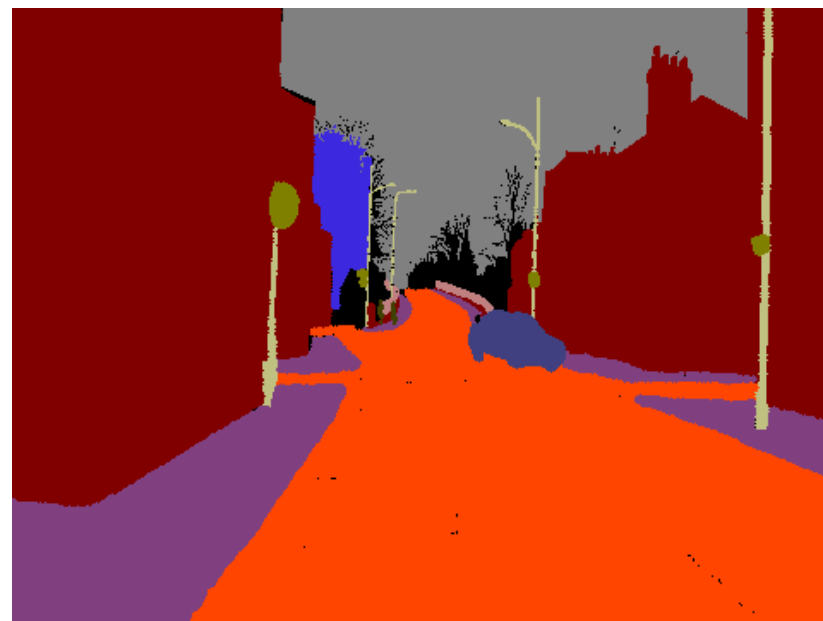


SegNet and Bayesian SegNet Tutorial
<<https://github.com/alexgkendall/SegNet-Tutorial>>

CamVidデータセットの例



input



label

比較するモデル(全5種類)

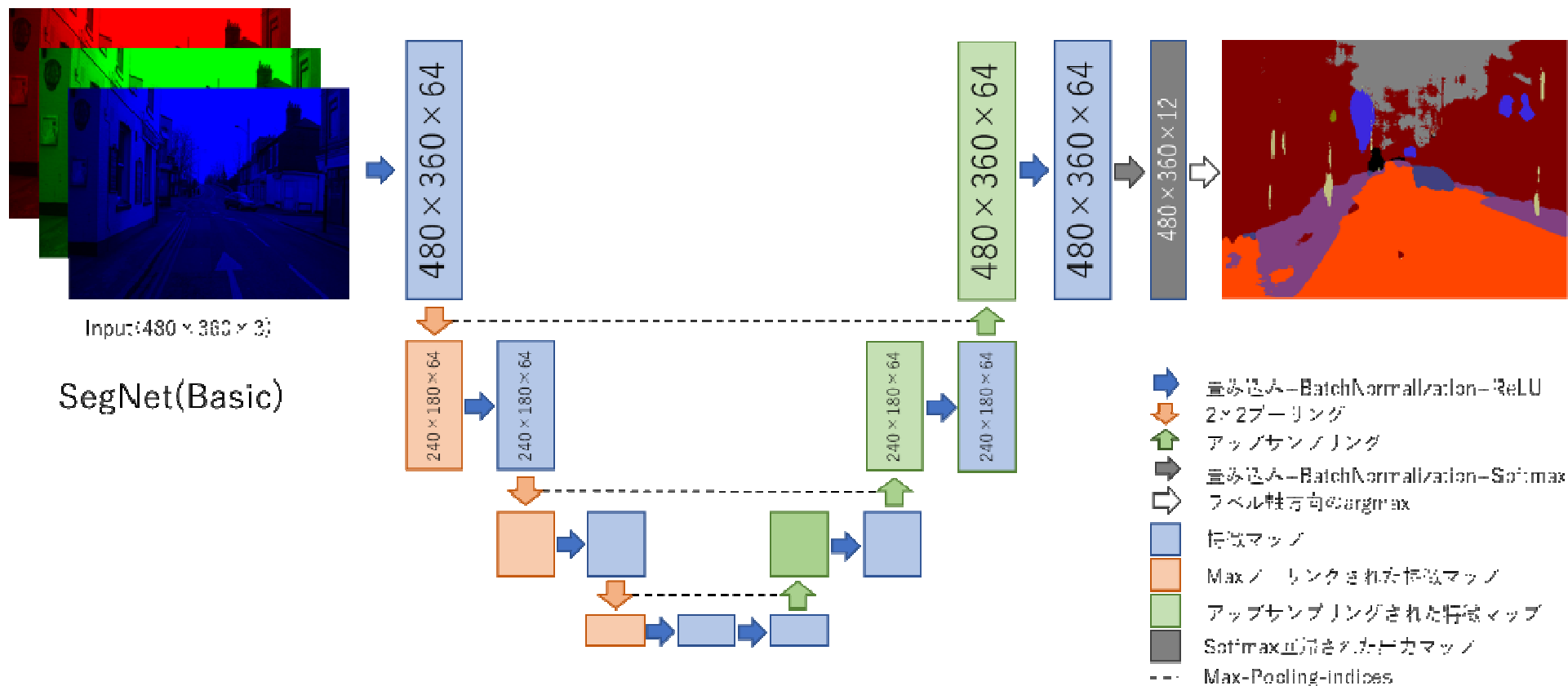
SegNetBasicの再現

- モデル構造やハイパーパラメータや、実験方法の基準とする



接続方法 接続箇所	Concatenate (既存手法)	Add (提案手法)
最大解像度のみ	USegTop	USegAddTop
全解像度	USegFull	USegAddFull

SegNetBasicの再現の図



モデル(SegNetBasicの再現)

SegNetBasicの再現

- 畳み込みカーネルの大きさ: 7×7
- チャンネルの数: 64
- 最大値プーリングの領域サイズ: 2×2

論文中では4回プーリングを行うとある

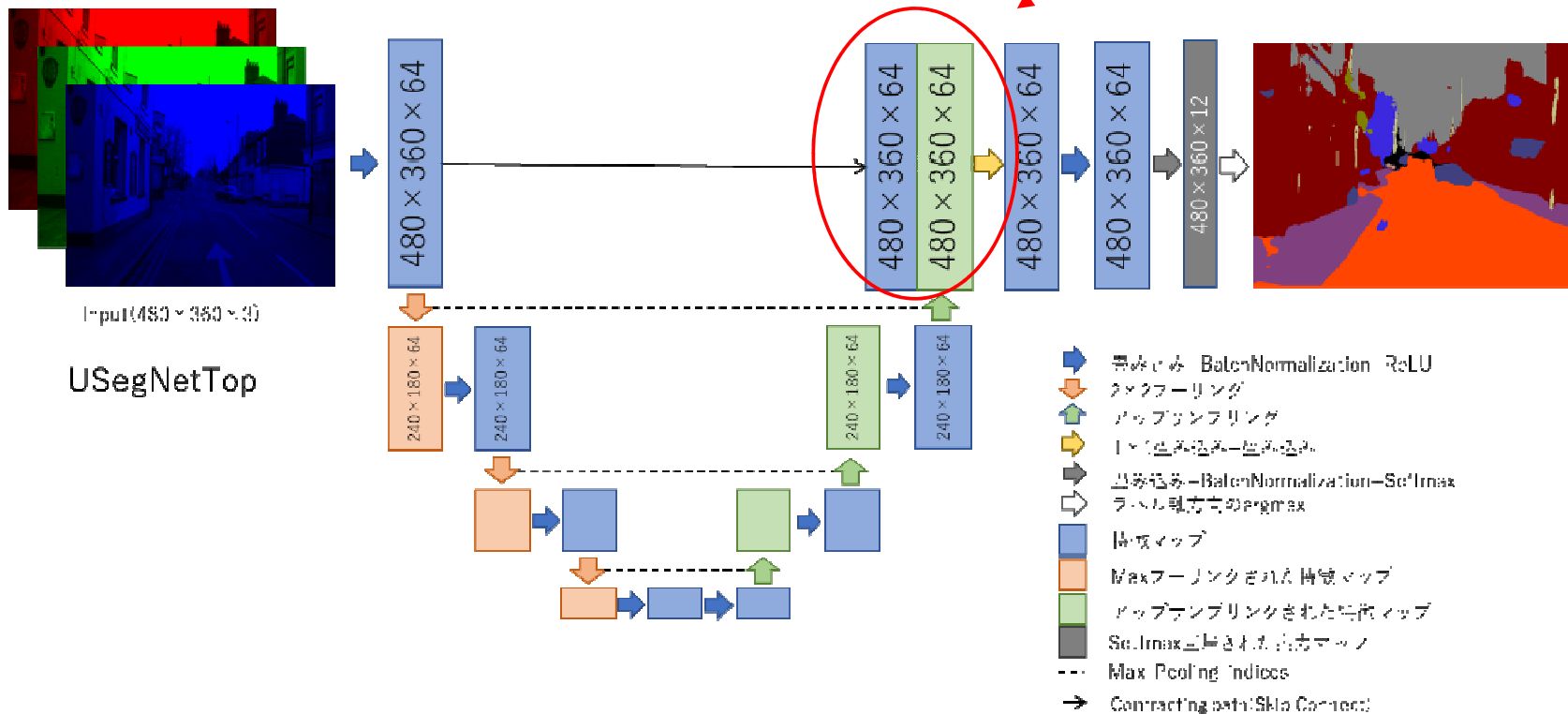
- 解像度的に不可能
- ソフトウェアの仕様？
- 今回は4回目のプーリングと1回目のアップサンプリングを削除

USegNetTop(既存手法)

SegNetBasicにSkip connectionを追加

- Concatenate結合と1×1畳み込みを追加
- 解像度が最大の組で結合

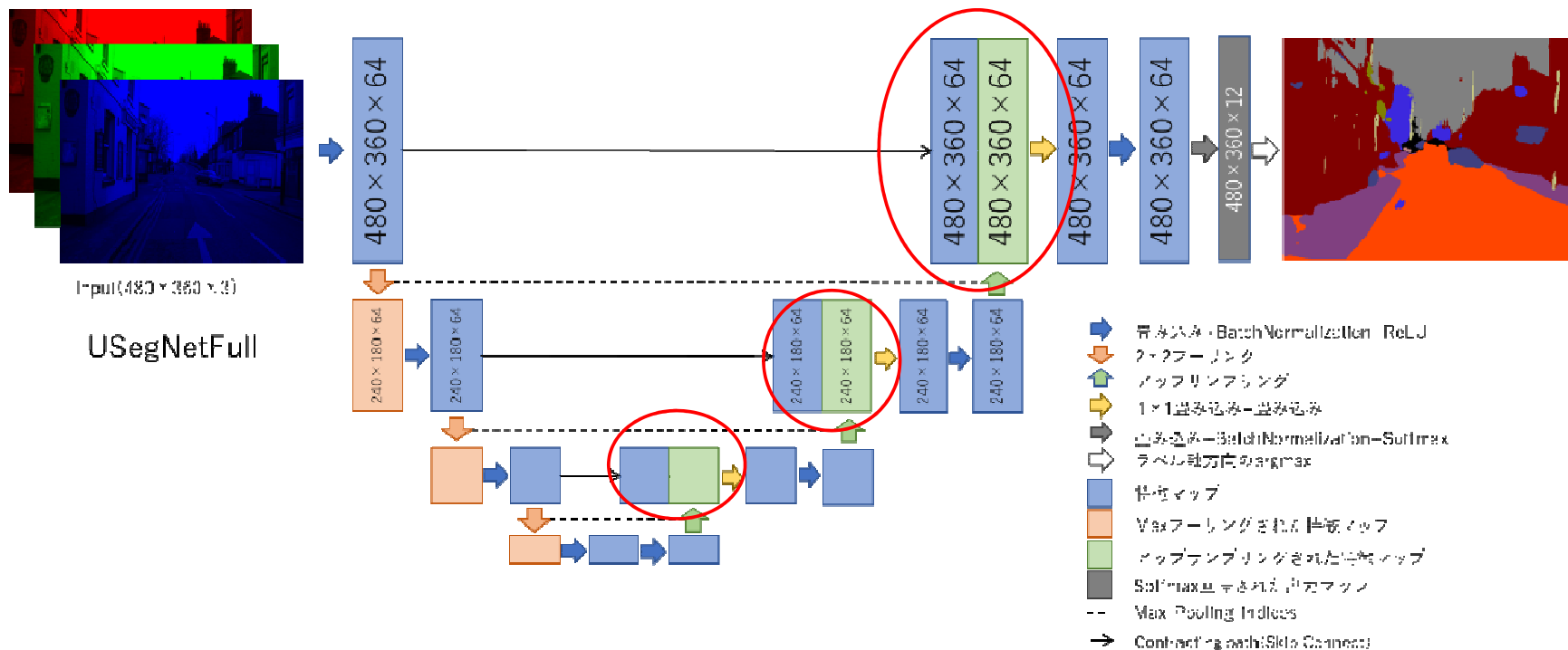
Concatenateされて
チャンネルの数が2倍
1×1畳み込み追加



USegNetFull(既存手法)

SegNetBasicにSkip connectionを追加

- Concatenate結合と1×1畳み込みを追加
- 全ての解像度の組で結合

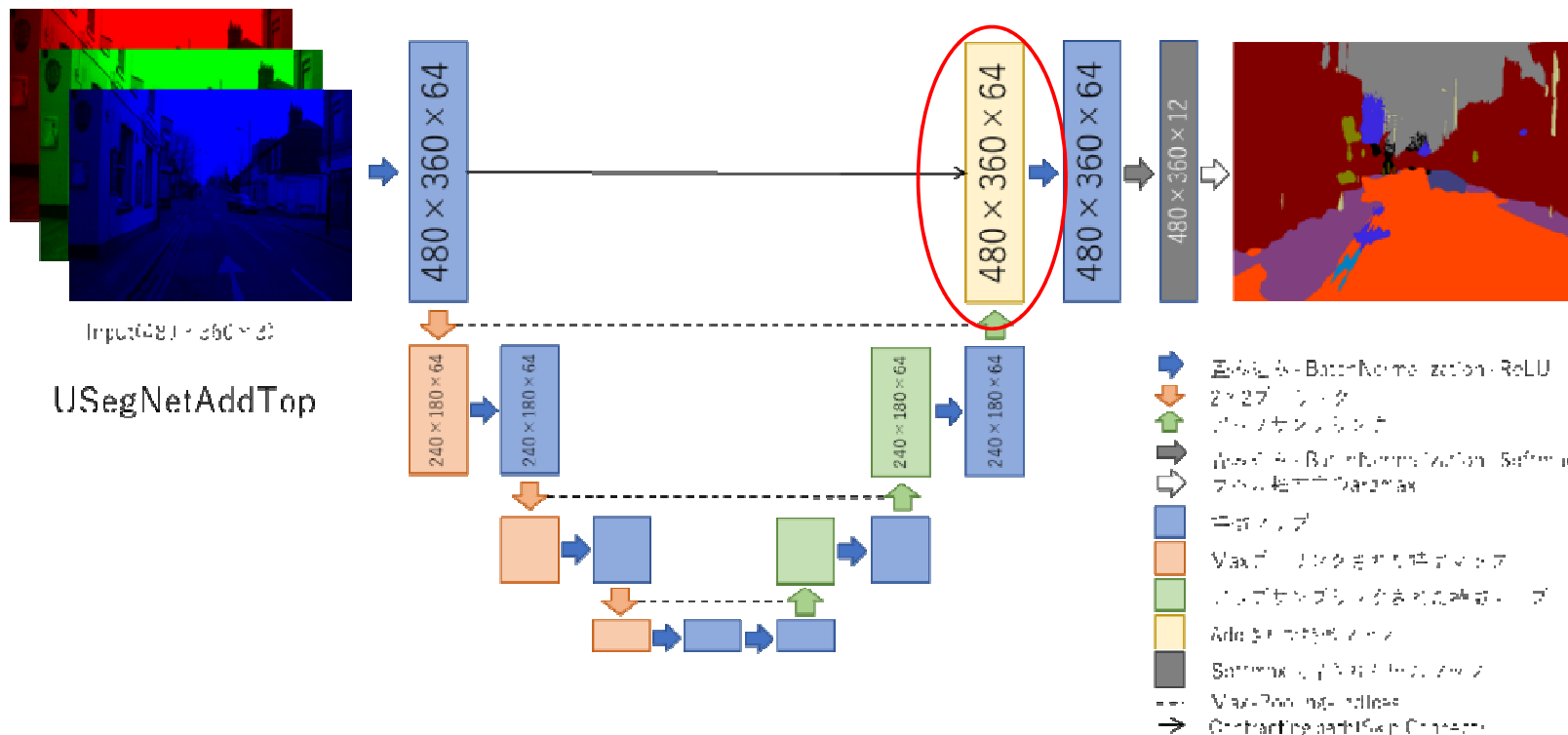


USegNetAddTop(提案手法)

USegNetTopの結合方法をAddに変更

1×1畳み込みを削除

Add結合
チャンネルの数変化なし



学習結果(予測性能指標)

	params	GA(WU)	CA(WU)	mIoU(WU)	WU=「Unlabeledを含めない」
Seg論文		0.84	0.546	0.463	Seg論文の値は 論文内に記載されていない 要素等があるため イタリックで表記
SegNet再現	1454064	0.806	0.497	0.410	
USegNetTop	1462256	0.843	0.545	0.461	
USegNetFull	1478640	0.833	0.510	0.431	
USegNetAddTop	1454064	0.840	0.535	0.458	
USegNetAddFull	1454064	0.834	0.578	0.469	

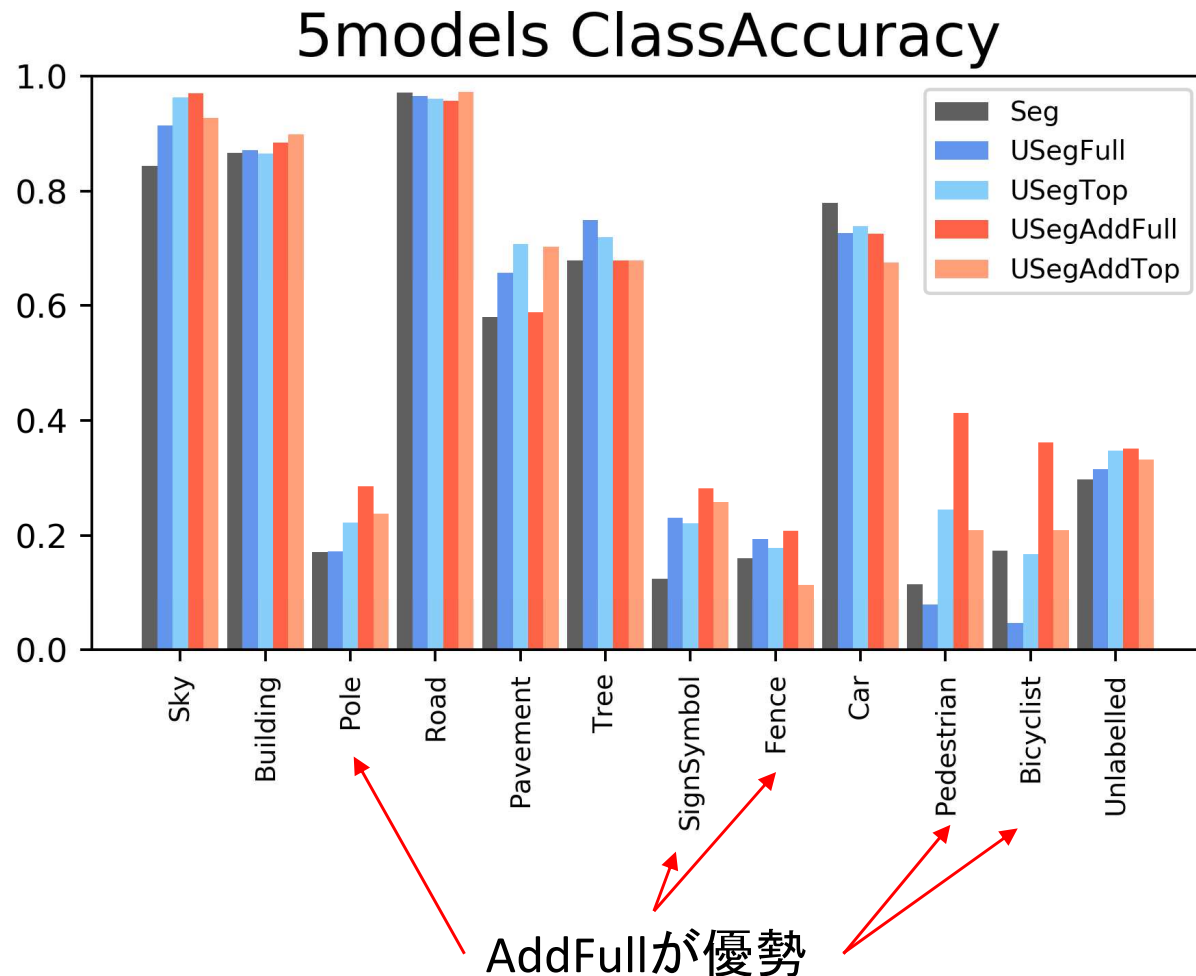
SegNetの再現具合がよくない→ソフト, ハードが原因?

USegNetTop: GAが最良

USegNetFull: すべての指標で良くない→パラメータの増加が原因?

USegNetAddFull: クラス別指標(CA, mIoU)が最良

クラス別性能指標(CA)

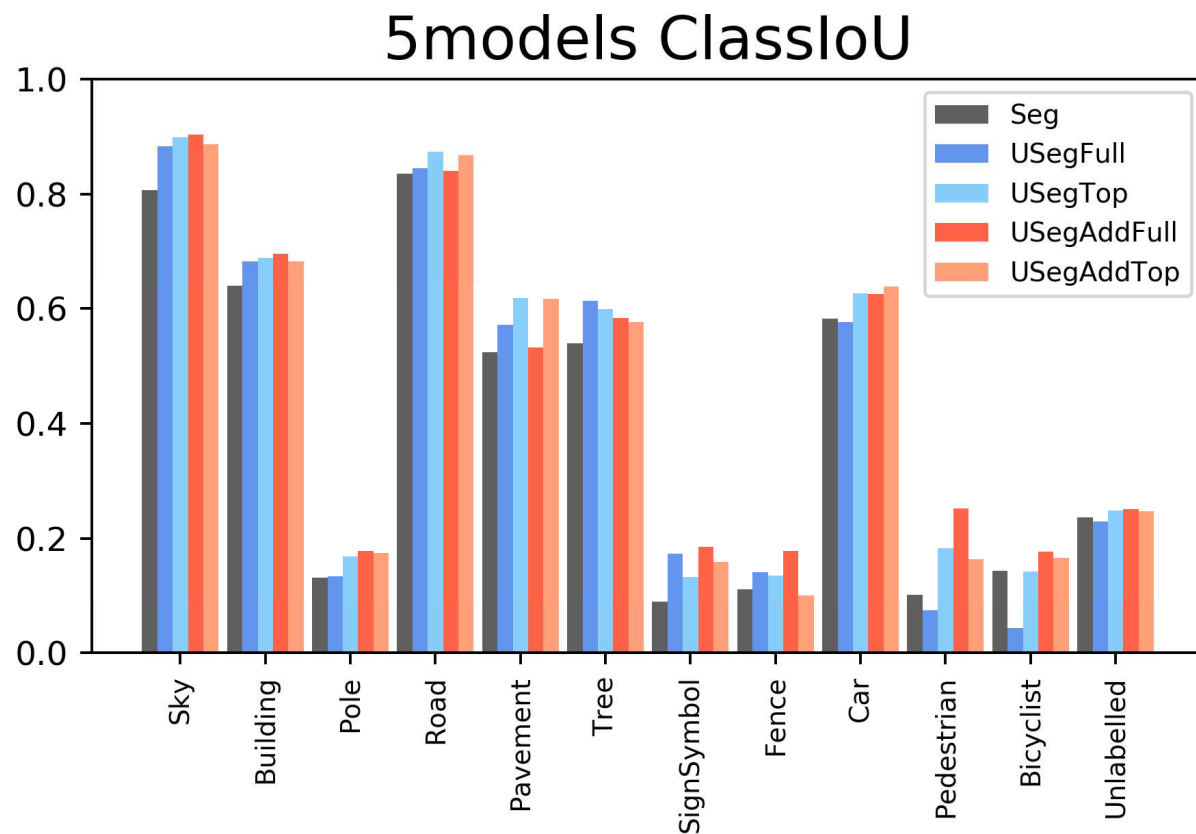


USegAddFullは
PedestrianやBicyclist
FenceやSignsymbol, Pole
で特に良い

RoadやPavementは上記の
ラベルに重なるように位置
するため少し悪い?

境界情報を捉え積極的に
予測しようとしたため
GAの性能が悪化したものの
CAでは優れた?

クラス別性能指標(mIoU)



USegAddFullが得意なもの、
苦手なものはCAとにている。

ただしUSegTopとの差が
縮まっている

mIoUは過剰な予測を行うと
悪化する指標



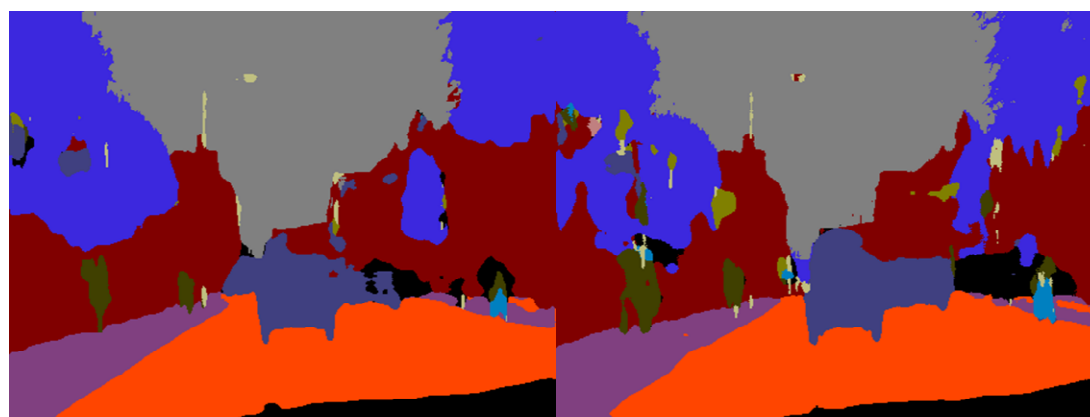
USegAddFullの積極性が
ノイズを捉えてしてしまい
mIoUは微増にとどまった?

予測結果の目視



input

Truth



USegTop

USegAddFull

USegTop

予測の輪郭が滑らか
PedestrianやByciclistの
予測は消極的

USegAddFull

予測の境界がギザギザ
PedestrianやByciclistの
予測は積極的

画面中央奥のUnlabeled
を予測してしまったり,
Poleのノイズを拾っている
(画面中央上部)

実験2:PASCAL VOC2012データセット

様々な画像分類用のラベル付きデータセット

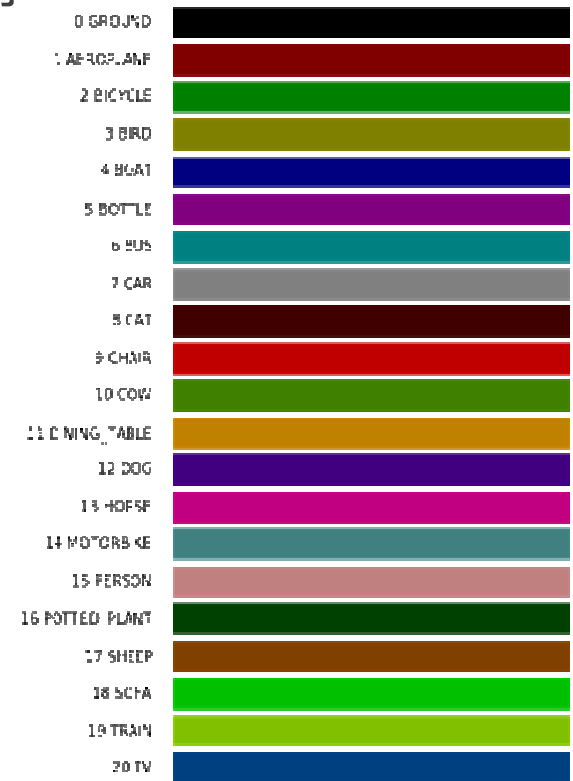
- セグメンテーション用が2913枚
- test:1525, validation:420,test:968の割合で使用
- ラベルの種類が20種+GROUNDと境界で2種
- 今回は境界をGROUNDに含め21種とする

解像度

- 様々なものが存在する
- 中心を224×224の大きさに切り抜いて用いる

CamVidとの違い

- シチュエーションやラベルが多様
- ほとんどがGROUNDである



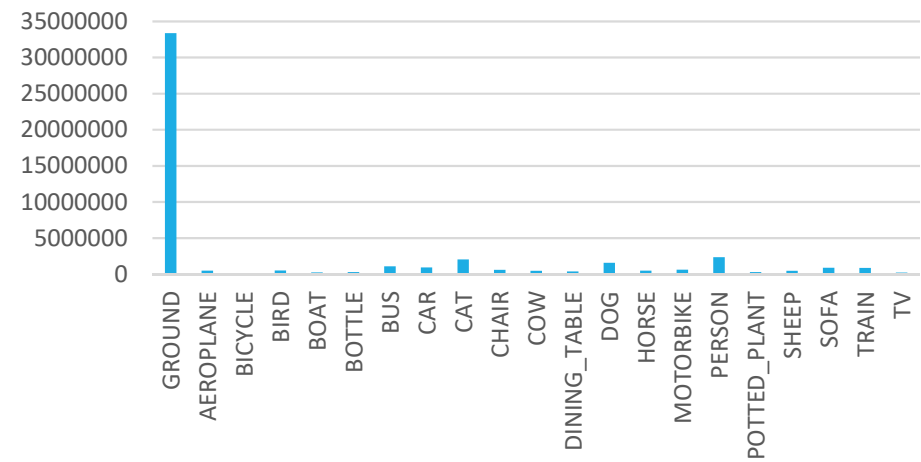
Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html>

PASCAL VOCの例



Inputとラベルの例

testセットのGTの数



予測性能指標

	GA(WG)	CA(WG)	mIoU(WG)	WG=「GROUNDを含めない」
Seg	0.4577	0.3884	0.2885	
USegTop	0.4853	0.4134	0.3112	
USegFull	0.4600	0.3773	0.2797	
USegAddTop	0.4661	0.3982	0.3056	
USegAddFull	0.4827	0.4202	0.3119	

CamVidデータセットと同様に

- USegTop→**GA**最良
- USegFull→すべてで悪い.
- USegAddFull→**CA, mIoU**最良

パラメータの数に関する考察

SegNetのtotal paramsが約2950万パラメータ

- Trainable paramsが1470万パラメータ

SegNetと比較して増加したtrainable paramsの数

- USegNetTop:約8000
- USegFull:約120万増加

VGG16の最大解像度

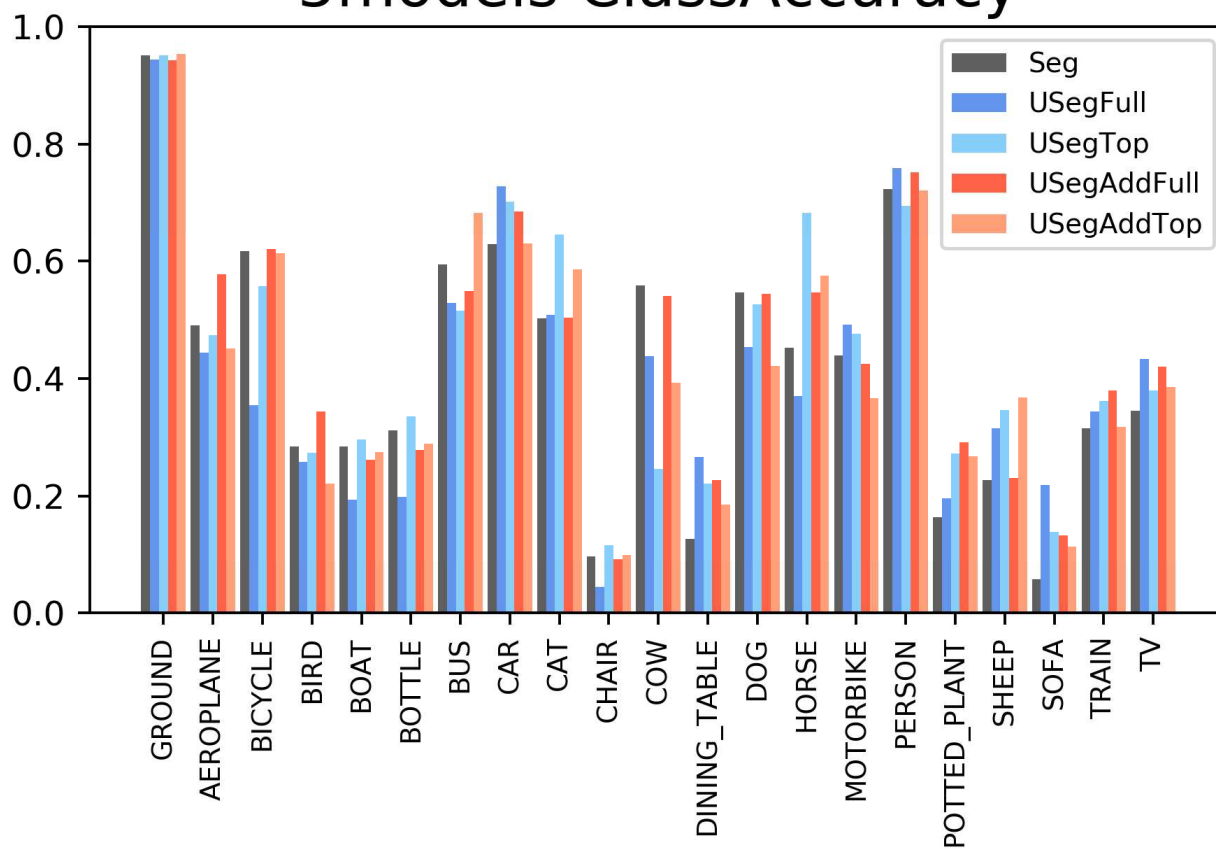
- 64チャンネル
- CamVidと同じ

- 層が深くなるにつれ
チャンネルが増加(最大512チャンネル)

	total params	trainable params
Seg	29,497,513	14,766,783
USegTop	29,505,769	14,775,039
USegFull	30,719,593	15,988,863
USegAddTop	29,497,513	14,766,783
USegAddFull	29,497,513	14,766,783

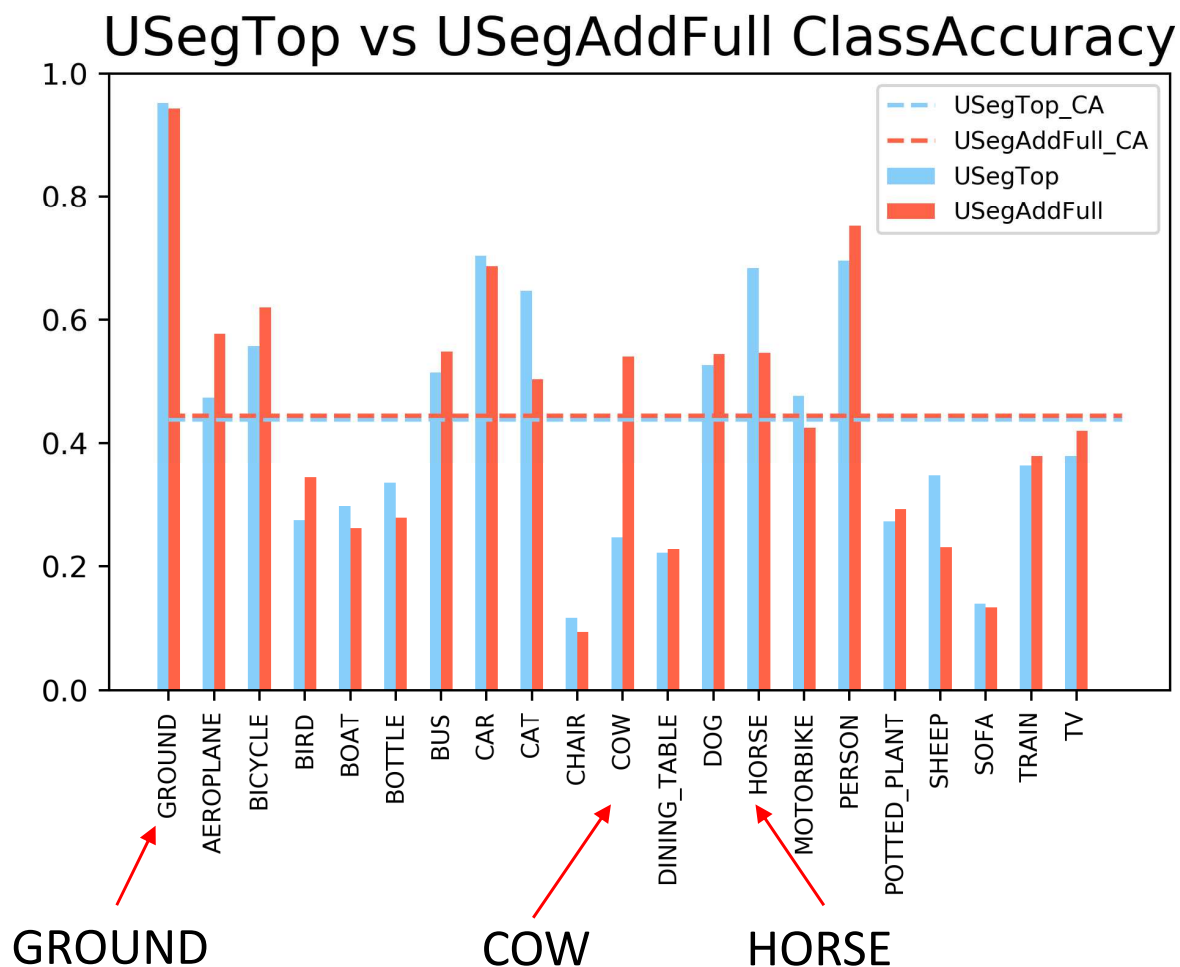
CA(5モデル)

5models ClassAccuracy



クラスが多いので
USegTopとUSegAddFullを見る

CA(USegTop vs USegAddFull)



USegTopはHORSEが高い
AddFullはCOWが高い
↓
偏って推定している？

同様の関係
CATとDOG
BICYCLEとMOTORBIKEなど

畳み込みや
プーリングサイズが小さいので
局所的予測のみしている？

GROUNDでAddFullが悪い
ノイズを拾っている？

COWとHORSEの例



USegTopは馬と予測しがち
AddFullは牛と予測しがち

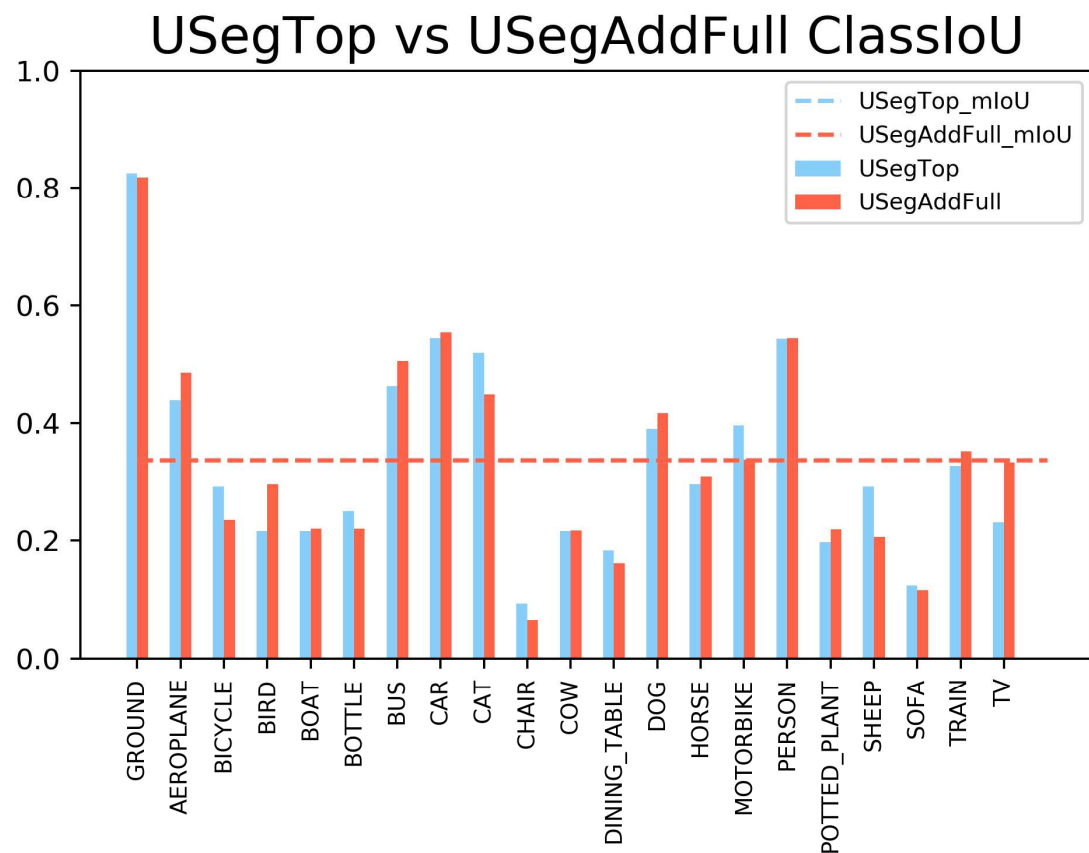
input

Truth

USegTop

USegAddFull

IoU(USegTop vs USegAddFull)



COWとHORSEの差が無くなった
mIoUが誤った推定に厳しいため？

CamVid

PedestrianやBicyclist
のような小さなモチーフの
周辺は道路や歩道がほとんど



共通した隣接情報が多く
境界情報を捉えやすかった？



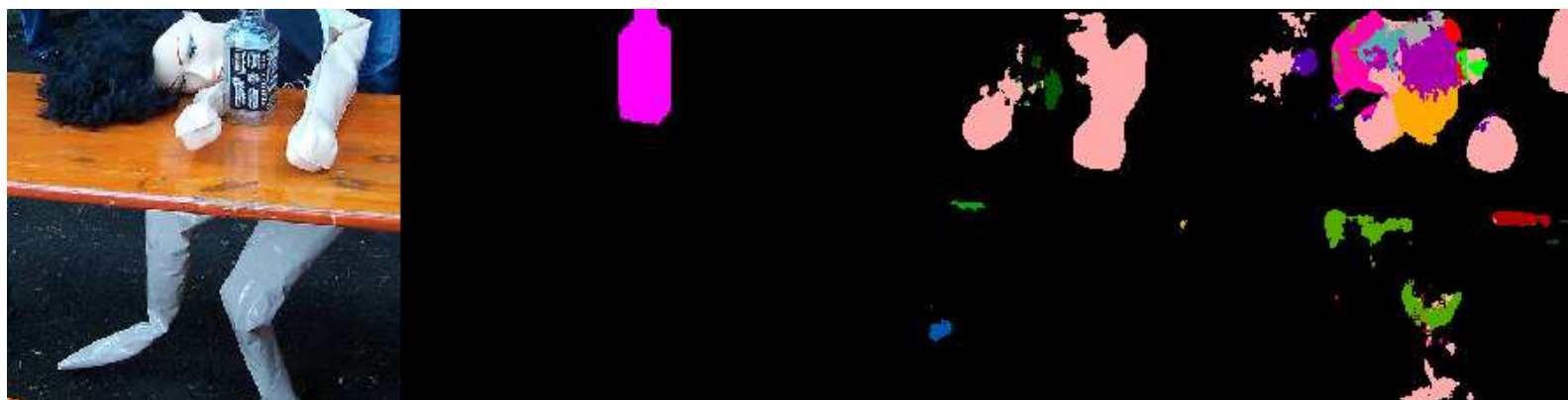
VOCは背景が様々
境界情報が捉えづらい？

VOCデータセットに対する考察

VOCはほとんどがGROUNDラベルである

- GROUNDにも類似したものが存在する
- USegAddFullは境界情報に過敏に反応する
- 背景の多いデータセットではノイズの分悪化している？

CamVidはUnlabeledが少なかったためUSegAddFullに合っていた？



input

Truth

USegTop

USegAddFull

目次

背景

事前知識共有

先行研究

提案手法

実験

結論・課題

実験の結論と課題

結論

- USegAddFull: パラメータを増やさず位置情報と低次元情報を伝達
- クラス別の分類指標を改善できる可能性がある
 - 細かな対象(歩行者や標識等)を推定することが得意
- しかしノイズを捉えすぎる可能性もある

課題

- 他のモデル構造や異なるハイパーパラメータで実験
- 様々な量, 質, 種類のデータセットでの検証
- 適度に境界情報を保持する手法の考案

参考文献1

[1]原田達也(2017), 画像認識, 講談社.

[2]光成滋生(2012), パターン認識と機械学習の学習 :
ベイズ理論に挫折しないための数学, 暗黒通信団.

[3]Bishop, C. M. (2006).

Pattern Recognition and Machine Learning. Springer-Verlag.

〔元田 浩/栗田多喜夫/樋口知之/松本祐治/村田 昇監訳(2012),

パターン認識と機械学習 上/下 ベイズ理論による統計的予測, 丸善出版.)

[4]太田満久/須藤広大/黒澤匠雅/小田大輔著(2018),

現場で使える!TensorFlow開発入門 kerasによる深層学習モデル構築手法, 翔泳社.

参考文献2

[5]Sergey Ioffe, Christian Szegedy(2015).

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

[6]Vijay Badrinarayanan, et al(2015).

SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561.

[7]Olaf Ronneberger, et al(2015).

U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597.

参考文献3

[8]Kaiming He, et al(2015).

Deep Residual Learning for Image Recognition.
arXiv:1512.03385

[9]Pulkit Kumar, et al(2018).

U-SEGNET: FULLY CONVOLUTIONAL NEURAL NETWORK
BASED AUTOMATED BRAIN TISSUE SEGMENTATION TOOL.
arXiv:1806.04429.

参考データセット

[10]Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. (2012).

The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

<<http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html>>

[11]Getting Started with SegNet

<http://mi.eng.cam.ac.uk/projects/segnet/tutorial.html>

[12]SegNet and Bayesian SegNet Tutorial

<<https://github.com/alexgkendall/SegNet-Tutorial>>

参考ライブラリ等

[13]Keras

<<https://keras.io/ja/>>

[14]Keras 「training lossがtesting lossよりもはるかに大きいのはなぜ？」

<<https://keras.io/ja/getting-started/faq/#training-losstesting-loss>>

[15]ykamikawa/tf-keras-SegNet

<<https://github.com/ykamikawa/tf-keras-SegNet>>

[16]Tensorflow

<<https://www.tensorflow.org/>>

参考ウェブサイト

[17]U-NetでPascal VOC 2012の画像をSemantic Segmentationする
<<https://qiita.com/tkktks10/items/0f551aea27d2f62ef708>>

[18]井崎武士 広がるディープラーニングの活用, NVIDIA JAPAN
◦ <https://www.slideshare.net/NVIDIAPJapan/ss-57361969>

[19] 多クラス交差エントロピー誤差関数とソフトマックス関数, その美しき微分
<<https://qiita.com/klis/items/4ad3032d02ff815e09e6>>

[20] 「画像認識をやってみました」の次にすること

<https://www.sigfoss.com/developer_blog/detail?actual_object_id=147>