論文紹介

Mask R-CNN

松永 葵

谷口研究室 B4

2019/04/17

目次

目次

物体検出について

- 物体検出
  - 物体を矩形領域で抽出
- セマンティックセグメンテーション
  - ピクセルひとつひとつにラベルを割り当てる
  - ただし、同じラベルの物体が重なっていると、物体同士の境界がわからない
- インスタンスセグメンテーション
  - 物体検出 + セマンティックセグメンテーション
  - それぞれの物体を区別しつつ、物体がある領域をピクセル単位で分類

目次

Mask R-CNN とは

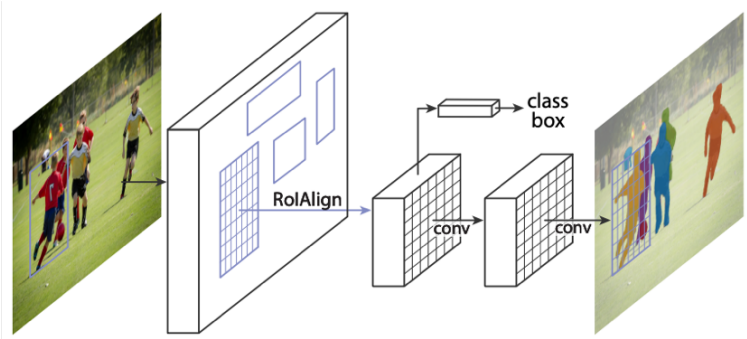

Figure: The Mask R-CNN framework for instance segmentation.

Mask R-CNN とは

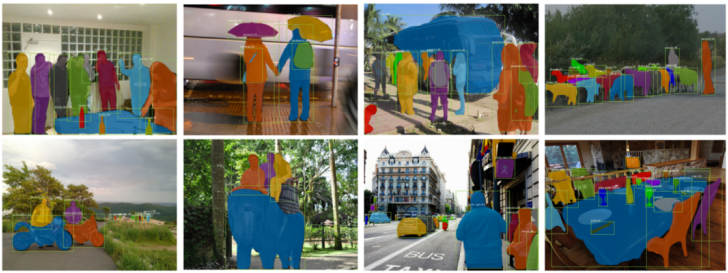

Figure 2: **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [4], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

Figure: Mask R-CNN results on the COCO test set.

従来手法との違い

- セグメンテーション優先戦略 (従来)
  - セマンティックセグメンテーション $\rightarrow$ 物体検出
  - ピクセルごとの分類から始めて、同じカテゴリのピクセルをインスタンスにカット
- インスタンス優先戦略 (Mask R-CNN)
  - 物体検出とセマンティックセグメンテーションを分離

目次

R-CNN (Regional with CNN features)

1. Region Proposal
   - Selective Search(物体らしさを見つける既存手法) を用いて、画像から RoI(物体候補領域) を探す
2. RoI を全て一定の大きさにリサイズして CNN にかけて features を抽出
3. 抽出した features を使って複数の SVM によって学習しカテゴリ識別 Regression によって bounding box を推定

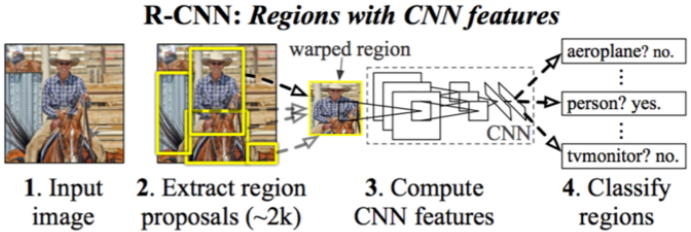

Figure: R-CNN : Region with CNN features

R-CNN (Regional with CNN features)

- 物体っぽい領域をたくさん見つけてきて、無理やりリサイズして CNN で特徴抽出、SVM でどのクラスか判定
- 欠点
  - 各項目ごとに別々に学習
  - 実行時間がめっちゃ遅い

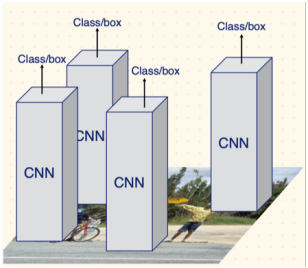

Figure: R-CNN

Fast R-CNN

- RoI pooling layer というシンプルな幅可変 pooling を行う
- Classification / bounding box regression を同時に学習させるための、multitask loss によって 1 回で学習ができるようにする
- オンラインで教師データを生成する工夫

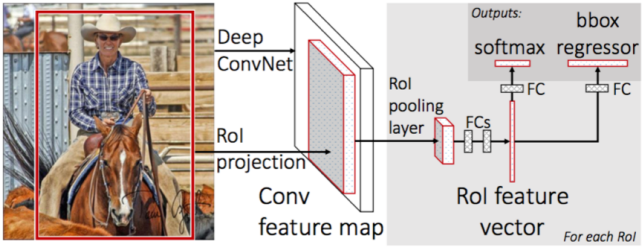

Figure: Fast R-CNN

Fast R-CNN

- R-CNN では、fine-tune/classification/bounding box regression をそれぞれ別々に学習する必要があったが、multi - task loss の導入により、Back-Propagation が全層に適用できるようになったため、全ての層の学習が可能となった (end to end ではない)

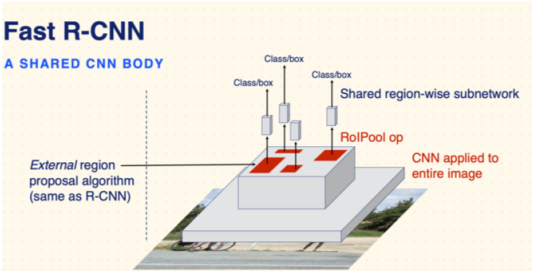

Figure: Fast R-CNN

Faster R-CNN

- Region Proposal Network (RPN)
  - end to end で学習可能
  - 物体候補領域を推定するネットワーク + RoI Pooling にクラス推定
    - ★ 物体かどうかを表すスコア (cls layer)
    - ★ 物体の領域 (reg layer)

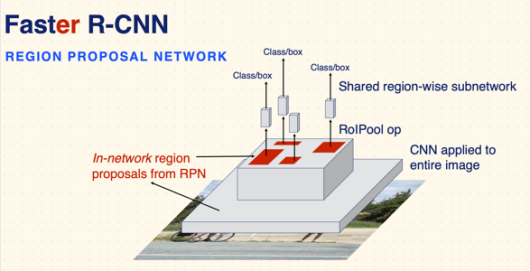

Figure: Faster R-CNN

Faster R-CNN

1. 画像全体の feature maps から予め決められた k 個の固定枠 (Anchor) を用いて特徴を抽出し、RPN の入力とする
2. 各場所について物体候補とすべきか推定
3. 物体候補として推定された出力枠 (reg layer) の範囲を、Fast R-CNN 同様 RoI Pooling し、クラスのネットワークの入力とすることで最終的な物体検出を実現

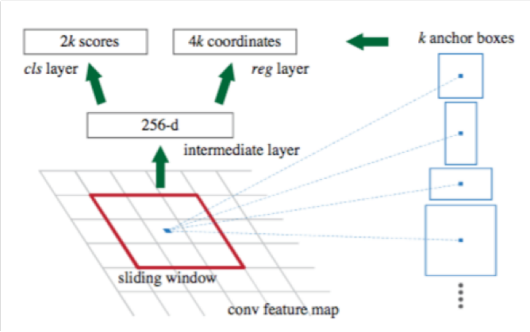

Figure: Faster R-CNN

目次

Faster R-CNN の拡張

- 既存の branch と並行して、mask branch を追加
- RoI のセグメンテーションマスクを予測
- mask と class の予測を切り離す
- RoI Pooling → RoI Align

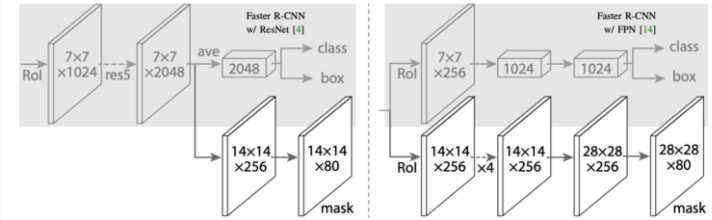

Figure: Head Architecture

multi task loss

$$L = L_{cls} + L_{box} + L_{mask}$$

(L : RoI ごとにサンプリングされた multi task loss)

- L_{cls} : 分類誤差
  - 物体カテゴリ数 + 1 クラス分類 (+1 は背景クラス)
  - 真のクラス u に対する事後確率 p^u の負の対数

$$L_{cls}(p, u) = -\log p^u$$

- L_{box} : 矩形回帰
  - 候補領域を真の bounding box に近づける回帰

$$L_{cls}(v, t) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i - v_i)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwize \end{cases}$$

multi task loss

- L_{mask}：マスク損失
  - ピクセルごとのシグモイドを適用し、平均バイナリクロスエントロピー損失として定義
    L_{mask} は k 番目の mask でのみ定義

$$L_{mask} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} [y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)]$$

($\hat{y}_{ij}^k$ は同じセルに対する k 番目の mask 予測)

→ マスクとクラスの予測を分離

	AP	AP_{50}	AP_{75}
softmax	24.8	44.1	25.1
sigmoid	**30.3**	**51.2**	**31.5**
	+5.5	*+7.1*	*+6.4*

Figure: Multinomial vs. Independent Masks (ResNet-50-C4)

(ちなみに)

- FCNs
  - ピクセルごとのマルチクラス分類
  - ソフトマックスと多項クロスエントロピー損失

  → セグメンテーションと分類を統合している
  - インスタンスセグメンテーションには向いていない

RoI Pooling

- ピクセル間の空間情報を維持するためのもの
- ある程度畳み込み処理を行った feature map から RoI を抽出し、あらかじめ定義されたサイズにスケーリング

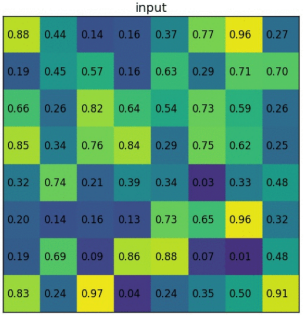

Figure: RoI Pooling

RoI Pooling

- 元画像の RoI を feature map に投影すると、サブピクセルレベルのずれが生じる
- RoI Pooling では、このずれを丸め込みながら Pooling を行う

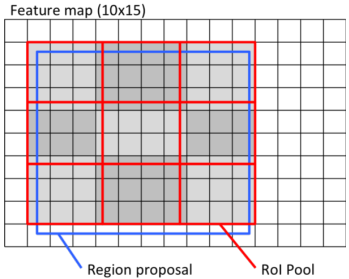

Figure: Feature map (10×15)

→ ピクセル精度の mask を予測するのに多大な悪影響

RoI Align

- RoI Pool の丸め込みを取り除き、抽出された特徴を入力と正しく位置合わせする

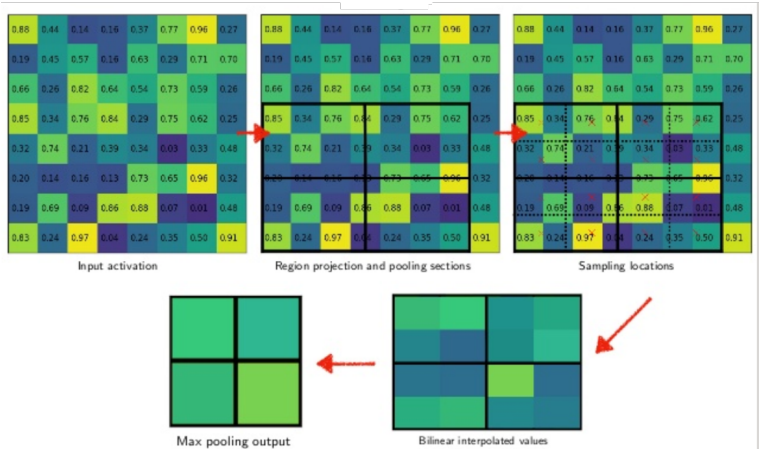

Figure: RoI Align

バイリニア補間

- 各セル内の 4 点の近傍 4 ピクセルからバイリニア補間 (双線形補間,bilinear interpolation) を用いて各点の値を計算する

↑ 周囲 4 画素の画素値の加重平均を計算

	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

Figure: RoI Align (ResNet-50-C5, stride 32)

目次

姿勢推定

- 人間の姿勢推定に容易に拡張が可能
  - キーポイントの位置を one-hot mask としてモデル化
  - K 個のキーポイントタイプ (左肩, 右肘など) ごとに一つずつ mask を予測

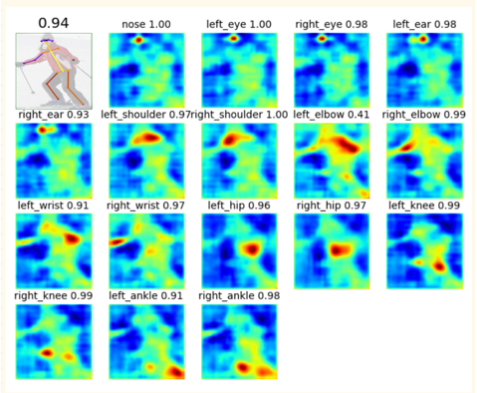

Figure: Keypoint の one-hot-mask

結果

Figure: Keypoint detection resluts on COCO (ResNet-50-FPN)

目次

まとめ

- インスタンスセグメンテーションのためのシンプルで効果的なフレームワーク
- Faster R-CNN を拡張したもの
- 他のタスクへの一般化が容易 (論文中では姿勢推定を紹介)

感想

- 今まで Faster R-CNN でやっていたものを Mask R-CNN でやるというよりは、セマンティックセグメンテーションで解決できなかったもの (同じラベルの物体が重なった時に境界がわからない) に対して使用するべき?

参考文献 I

- Kaiming He, “Mask R-CNN,” in International Conference on Computer Vision (ICCV), 2017.
- R. Girshick, “Fast R-CNN”, in International Conference on Computer Vision(ICCV), 2015.
- J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation”, In Conputer Vision and Pattern Recognition (CVPR), 2016.
- http://on-demand.gputechconf.com/gtc/2017/presentation/s7783-ross-girshick-fast-unified-method-object-detection-instance-segmentation-human-pose-estimation.pdf (2019/4/16)
- 物体検出についての歴史まとめ https://qiita.com/mshinoda88/items/9770ee671ea27f2c81a9 (2019/4/13)

参考文献 II

最新の物体検出手法 Mask R-CNN の RoI Align と Fast(er) R-CNN の RoI Pooling の違いを正しく理解する
https://qiita.com/yu4u/items/5cbe9db166a5d72f9eb8
(2019/4/14)