

102. 深層学習の適用方法 (物体検知)

秋葉洋哉

2024 年 7 月 15 日

1 物体検知

1.1 概要

深層学習を用いた物体検知の研究の流れは、徐々に難易度を上げてきた。その流れは以下のとおりである。

- 分類 (Classification) : 画像に対して複数のクラスラベルを出力
- 物体検知 (Object Detection) : 画像中の物体の位置を推定 (Bounding Box)
- 意味領域分割 (Semantic Segmentation) : 各ピクセルに対して単一のクラスラベルを出力
- 個体領域分割 (Instance Segmentation) : 各ピクセルかつ個体毎に対して単一のクラスラベルを出力




物体検出タスクの種類			
	目的	特徴	出力イメージ
物体検出	画像内の物体を検出し、それぞれの物体に対して境界ボックスを指定する	物体の位置とカテゴリを特定する。各物体が独立して検出される。	
セマンティックセグメンテーション	画像内の全てのピクセルをカテゴリに分類する。	同じカテゴリの物体でも個々に区別せず、全てのピクセルが何らかのカテゴリに分類される。	
パノプティックセグメンテーション	セマンティックセグメンテーションと物体検出の組み合わせで、画像内の全ての物体を個別に認識し、それぞれのピクセルにカテゴリを割り当てる。	個々の物体を識別しつつ、画像内の全ピクセルをカバーする。	

図 1: 物体検出タスクの種類

物体検出では、BoundingBox と Confidence の 2 種が同時に出力される。BoundingBox は物体の位置を示し、Confidence はその物体が存在する確信度を示す。

1.2 データセット

物体検知に用いられるデータセットは、VOC12, ILSVRC17, MS COCO18, OICOD18 などがある。

VOC12(Visual Object Classes) は 20 クラス、Train+Val は 11,540 枚、Box/画像 (1 画像当たりの物体数)=2.4 であり、(PASCAL VOC Detection Challenge) というコンペティションで使用されていた。VOC12 の Ground Truth の BB に関する情報は $\langle x_{min} \rangle$, $\langle y_{min} \rangle$, $\langle x_{max} \rangle$, $\langle y_{max} \rangle$ の四つのタグが示す長方形で与えられる。

ILSVRC17(ImageNet Large Scale Visual Recognition Challenge) は 200 クラス、Train+Val は 476,668 枚、Box/画像 (1 画像当たりの物体数)=1.1 であり、(ImageNet Large Scale Visual Recognition Challenge) というコンペティションで使用されていた。ImageNet は、画像認識のためのデータセットであり、ILSVRC はそのサブセットである。

MS COCO18(Microsoft Common Objects in Context) は 80 クラス、Train+Val は 123,287 枚、Box/画像 (1 画像当たりの物体数)=7.3 であり、(Microsoft Common Objects in Context) というコンペティションで使用されていた。

OICOD18(Open Images Challenge Detection) は 500 クラス、Train+Val は 1,743,042 枚、Box/画像 (1 画像当たりの物体数)=7.0 であり、(Open Images Challenge Detection) というコンペティションで使用されていた。Open Images V4 は、6000 クラス以上/900 万枚以上の画像を含むデータセットであり、そのサブセットが OICOD18 である。

Box/画像 (1 画像当たりの物体数) は、小さいほどアイコン的な映りであり、日常感とは異なる。大きいほど日常感に近い映りであり部分的な重なりも見られるようになる。

どのような物体検知モデルを作成したいかを考える時は、Box/画像 (1 画像当たりの物体数) とクラス数を考慮する必要がある (図 2)。

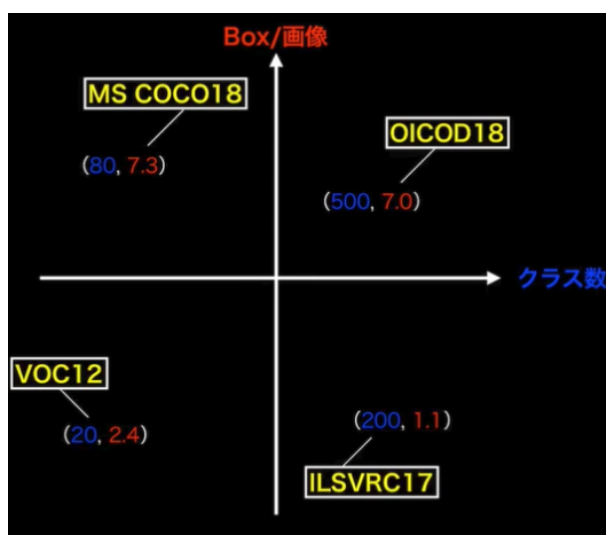


図 2: 代表的データセットのポジショニングマップ

1.3 評価指標

混合行列 (図 3) を確認すると、一般的に予測する際の評価指標は、精度 (Precision), 再現率 (Recall), 正解率 (Accuracy) が用いられる。これらの指標は、Threshold によって変化する。クラス分類では、Threshold に応じて、分類されるクラスが変化するが、物体検出では、Threshold に応じて、BoundingBox の数が変化する (図 4)。

		予測		
		Positive	Negative	
実際	Positive	True Positive	False Negative	再現率 (Recall) $\frac{TP}{TP+FN}$
	Negative	False Positive	True Negative	
		適合率 (Precision) $\frac{TP}{TP+FP}$	正解率 (Accuracy) $\frac{TP+TN}{TP+FP+FN+TN}$	

図 3: 混合行列 (復習)

閾値変化に対する振る舞い									
クラス分類					物体検出				
	conf.	pred.				conf.	pred.	BB	
S1	0.88	1	S1	0.88	1	P1	0.92	人	x1, y1 w1, h1
S2	0.82	1	S2	0.82	1	P2	0.85	車	x2, y2 w2, h2
S3	0.71	1	S3	0.71	0	P3	0.81	車	x3, y3 w3, h3
S4	0.52	1	S4	0.52	0	P4	0.70	犬	x4, y4 w4, h4
S5	0.49	0	S5	0.49	0	P5	0.69	人	x5, y5 w5, h5
S6	0.44	0	S6	0.44	0	P6	0.54	車	x6, y6 w6, h2
Threshold 0.5					Threshold 0.5				
Threshold 0.8					Threshold 0.8				
					Ground-Truth				
									

図 4: Threshold の変化に対するクラス分類と物体検出の違い

物体検知において精度は、予測で出力した BBox の数に対する、正解の BBox の数の割合であり、予測の BBox が真の BBox 数よりも多い場合、精度の高い順に TP とみなし、残りは FP とみなすことで求まる。

つまり、予測の BBox が多いほど精度は下がる。一方再現率は、真の BBox の数に対する、予測で出力した BBox の数の割合であり、予測の BBox 数が真の BBox 数よりも少ない場合、その分を FN とみなすことで求める。つまり、予測の BBox が少ないほど再現率は下がる。

この Precision-Recall を Threshold の関数としてプロットしたものを Precision-Recall curve と呼ぶ。また、Precision-Recall curve の下の面積を求めたものを AP(Average Precision) と呼ぶ。AP は、物体検知の性能を評価する指標であり、AP が高いほど性能が良いとされる。

物体検知の評価指標には、AP(平均適合率) と mAP(平均適合率) がある。AP は、クラスごとの AP の平均値であり、mAP は、全クラスの AP の平均値である。mAP は、物体検知の性能を評価する指標であり、mAP が高いほど性能が良いとされる (図 5)。

【計算例】		車	人	犬
	AP	86.4	67.2	64.3
	mAP	約72.6		

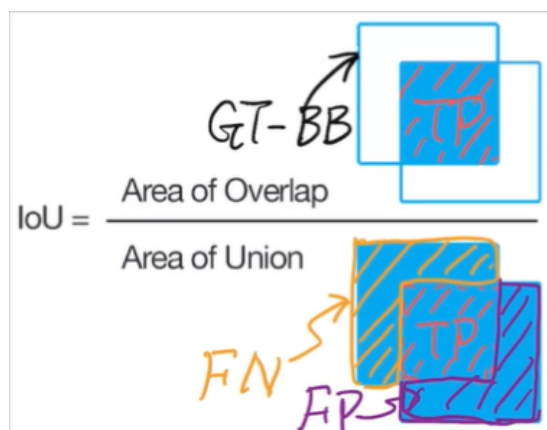
図 5: AP と mAP の違い

IoU(Intersection over Union) は、真の BoundingBox と予測の BoundingBox の重なり具合を示す指標であり、IoU が 0.5 以上の BoundingBox を正解とする場合が多い。IoU は以下で定義される。

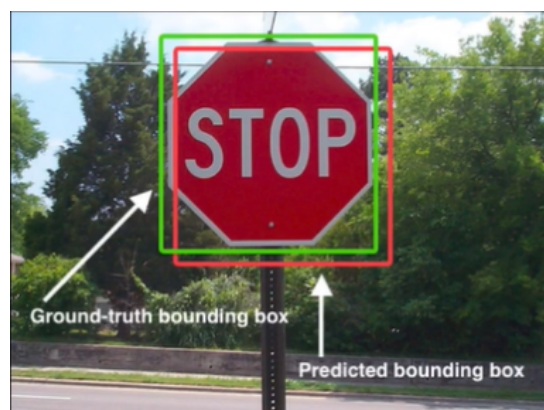
$$\text{IoU} = \frac{(\text{Area of Overlap})}{(\text{Area of Union})} \quad (1)$$

$$= \frac{TP}{TP + FP + FN} \quad (2)$$

これを図で示したのが図 6a, 図 6b である。物体検知では、Confidence と IoU の 2 つの指標を用いて評価することが多い。



(a) IoU の定義



(b) IoU の例

図 6: IoU について

1.4 検出速度

物体検知のモデルの性能を評価する際には、検出速度も重要な指標である。検出速度は、FPS(Frame Per Second) で表され、1 秒間に何枚の画像を処理できるかを示す。検出速度は、モデルの複雑さや画像サイズによって変化する。検出速度が速いほど、リアルタイムでの物体検知に適していると言える。横軸に FPS, 縦軸に mAP を取ったグラフや、横軸に inference time, 縦軸に mAP を取ったグラフを作成することで、検出速度と性能のトレードオフを可視化することができる (図 7)。

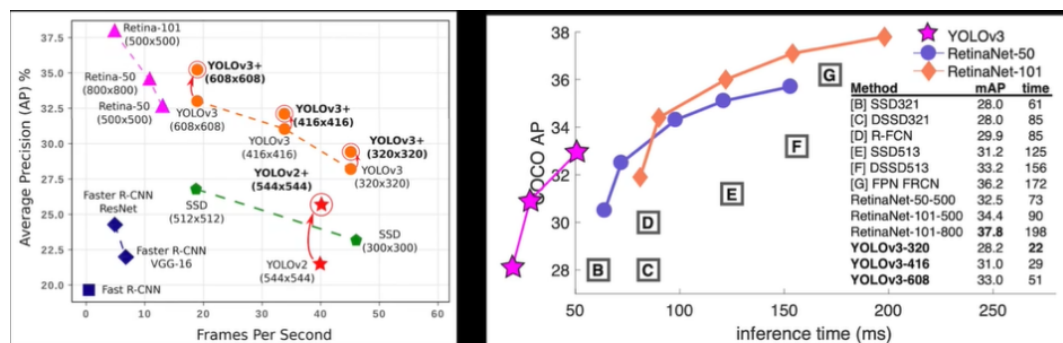


図 7: 検出速度と性能のトレードオフ

2 深層学習以降の物体検知

物体検知のフレームワークは深層学習の進化に伴って同時に進化した。メカニズムに基づき大きく分けて2つに分類できる。

- 2段階検出 (2-stage detection) : 物体の候補領域を検出し、その領域に対して物体のクラスと位置を別々に推定する (相対的に精度が高いが、計算量が多く、推論も遅い)
- 1段階検出 (1-stage detection) : 物体のクラスと位置を同時に推定する (相対的に精度が低いが、計算量が少なく、推論も速い)

2段階検出は、RCNN, FastRCNN, FasterRCNN, MaskRCNN などがある。1段階検出は、SSD, YOLO, RetinaNet, DetecorNet, CornerNet などがある。リアルタイムで検出する場合は、1段階検出が適している。

3 SSD(Single Shot MultiBox Detector)

SSD は、1 段階検出の代表的なモデルであり、物体のクラスと位置を同時に推定する。その方法は、以下の通りである。

1. 画像を入力する。
2. 画像上に適当に Default Box を配置する。
3. Default Box を変形し、検出したい物体のクラスと Conf. を推定する。

SSD は VGG16 というネットワークをベースにしており、VGG16 の最後の畳み込み層を利用して特徴マップを取得する (VGG16 の 16 は Convolution 層の数である)。

3.1 SSD のアーキテクチャ

SSD のアーキテクチャは、以下の通りである。

1. 入力：画像を入力。入力サイズに応じて SSD300, SSD512 などと表記される。
2. 隠れ層：VGG16 の 10 層目に対応する Conv 層 (38×38 の Conv 層)、や、VGG16 の FC(Fully Connected) 層 2 層に対応する層 (FC6, FC7 層) を Conv 層に変更して、様々なサイズの特徴マップを取得する。
3. 出力：Default Box(クラス数 + 4(場所に関する項)) を出力する。1 つの特徴マップに対して、 k 個の Default Box を出力する ($k \times (\text{クラス数} + 4)$) ため、特徴マップのサイズが $m \times n$ の場合合計で、($k \times (\text{クラス数} + 4)$) mn 個の Default Box を配置する。

特徴マップの解像度が高いほど、小さな物体を検出することができ、特徴マップの解像度が低いほど、大きな物体を検出することができるため、様々な解像度の特徴マップを用いることで、様々なサイズの物体を検出することができる。

VOC データセットでは、クラス数 20 に背景クラスが追加されるため、クラス数 21 に対して、合計の出力は、 $8,732 \times (21+4)$ 個となる。

このように多数の Default Box を用意したことで、1 つの物体しか映っていないくても、複数の Predicted-BBox が表示されることになってしまう。この問題を解決するために、Non-Maximum Suppression(NMS) という手法が用いられる。NMS では、Predicted-BBox 同士の IoU を計算し、閾値を超えた場合 1 つを残して他を削除することで、1 つの物体に対して 1 つの Predicted-BBox を残す。

また、多数の Default Box を用意することで、背景を示す Predicted-BBox に対して、物体を示す Predicted-BBox の比率がとて小さくなってしまう。この問題を解決するために、Hard Negative Mining という手法が用いられる。Hard Negative Mining では、背景を示す Predicted-BBox と、物体を示す Predicted-BBox の比率が 1:1 3:1 までになるように制約を課すことで、背景を示す Predicted-BBox の比率を減らす。

3.2 損失関数

SSD の損失関数は、以下の通りである。

$$L = L_{\text{loc}} + L_{\text{conf}} + L_{\text{hardneg}} \quad (3)$$

$$L_{\text{loc}} = \sum_i^N \sum_{j \in Pos}^N \text{SmoothL1}(l_{ij} - \hat{g}_{ij}) \quad (4)$$

$$L_{\text{conf}} = - \sum_i^N \log(\hat{c}_{ij}) \quad (5)$$

$$L_{\text{hardneg}} = - \sum_i^N \log(\hat{c}_{ij}) \quad (6)$$

ここで、 L_{loc} は、位置に関する損失関数、 L_{conf} は、クラスに関する損失関数、 L_{hardneg} は、背景を示す Predicted-BBox に対する損失関数である。また、 l_{ij} は、Predicted-BBox の位置、 \hat{g}_{ij} は、Ground Truth の位置、 \hat{c}_{ij} は、Predicted-BBox のクラス、 Pos は、Positive Sample を示す。

4 Semantic Segmentation

4.1 概要

Semantic Segmentation は、画像の各ピクセルに対して単一のクラスラベルを出力するタスクである。画像のピクセルごとにクラスを推定するため、画像全体に対してクラスラベルを付与することができる。

Semantic Segmentation は、FCN(Fully Convolutional Network) や U-Net, SegNet, DeepLab などのモデルが提案されている。これらのモデルは、畳み込みニューラルネットワークを用いて画像の各ピクセルに対してクラスラベルを推定する。

ネットワーク内でクラスを分類するためには、解像度を落とす必要がある。一方で、各ピクセル毎のクラスを求めるためには、出力時に元の画像の解像度に戻さなければならない。そのため、画像の解像度を落としてから、再び元の解像度に戻す、Up-sampling という処理が必要となる。

4.2 FCN(Fully Convolutional Network)

FCN は、Semantic Segmentation のアーキテクチャモデルの一つであり、VGG16 における最後の FC 層をすべて Conv. 層に置き換え、すべての層が畳み込み層で構成されたネットワークである。このような変換を用いることで、出力としてヒートマップのようなものが得られることになる。

Semantic Segmentation では、最終的に元画像と同一の解像度の画像を出力するため、FCN によって全結合層を畳み込み層に変換することで、Up-Sampling 前の無駄な処理を省くことができる。

FCN の特徴として、最終的に全結合層を有さないため、入力画像の解像度は任意のサイズであっても良い。

— Pooling の重要性 —

Semantic Segmentation において、画像の解像度をもとに戻すために、Up-sampling が必要ということが分かったが、そもそも Pooling しなければ Up-sampling も必要ないと思うかもしれない。

まず、機械が正しく、犬であるか、猫であるかを認識するためには、受容野にある程度の大きさが必要である。受容野を広げる例としては、深い Conv. 層を用いる、Pooling+Stride を行う、Dilated Convolution などが挙げられる。

1 つ目の深い Conv. 層を用いる方法は、受容野を広げることができるが、計算量が多くなるという問題がある。2 つ目の Pooling+Stride を行う方法は、計算量を多くせずに、受容野を広げることができる。

よって、Pooling は、受容野を広げ、その画像に対する特徴を抽出するために重要であるので、欠かすことができない処理になる。

3 つ目の Dilated Convolution は、後ほど詳しく説明する。

4.3 Up-sampling

Up-sampling は、画像の解像度を元に戻す処理であり、画像の解像度を落とす Pooling とは逆の処理である。Up-sampling の一つとして、Deconvolution や Transposed convolution という仕組みが用いられる。

4.3.1 Deconvolution

Deconvolution は、以下の手順で行われる。

1. Kernel size, padding, stride の各値を指定する
2. 特徴マップの pixel 間隔を stride だけ空ける
3. 特徴マップの周りに、 $(\text{kernel size} - 1) - \text{padding}$ だけ余白を作る
4. 畳み込み演算を行う

この手順を示した画像が、図 8 である。この手法は、解像度を基に戻すことができるが、Pooling で失われた情

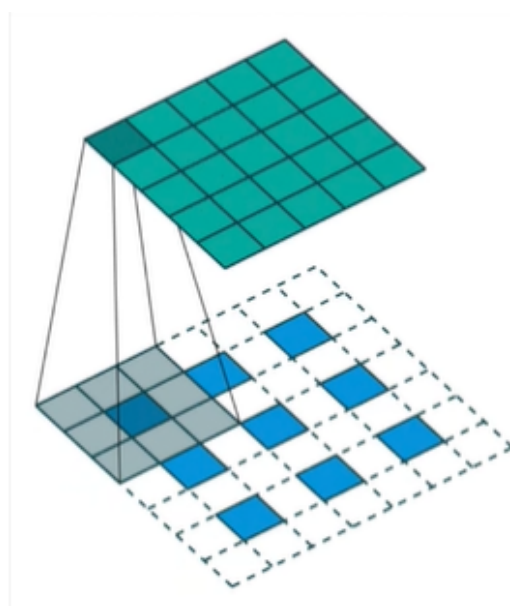


図 8: Up-sampling の概要

報が復元されるわけではない。Pooling によって失われた輪郭のようなローカルな情報は、Skip Connection を用いて、解像度の高い Pooling 層と結合することで、復元していくことになる (図 9)。

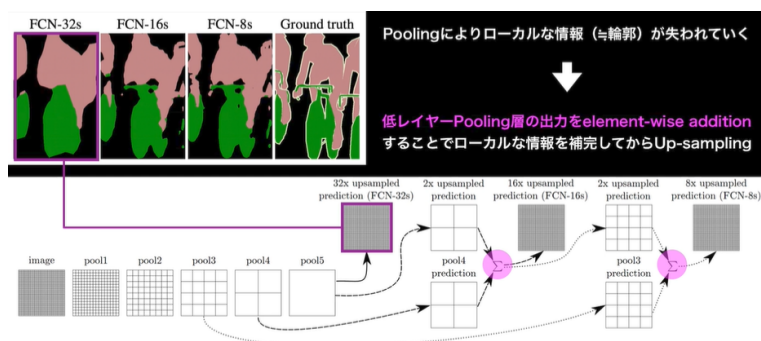


図 9: Up-sampling と Skip Connection の結合

4.3.2 Unpooling

Unpooling は、Pooling で画像の解像度を落とした際に失われた情報を復元するための処理である。Unpooling は、Pooling 時に記憶しておいた位置情報を用いて、元の画像の解像度に戻す処理である。Unpooling の手法として、以下の 2 つがある。

- Max Unpooling : Pooling 時に Max 値を取った位置を記憶しておき、その位置に Max 値を配置する
- Nearest Unpooling : Pooling 時に Max 値を取った位置を記憶しておき、その位置に最も近い値を配置する

Max Unpooling は、Pooling 時に Max 値を取った位置を記憶しておくため、元の画像の解像度に戻す際に、Max 値を配置する。一方、Nearest Unpooling は、Pooling 時に Max 値を取った位置を記憶しておくため、元の画像の解像度に戻す際に、その位置に最も近い値を配置する。

4.4 U-Net

U-Net は、Semantic Segmentation のモデルの一つであり、Encoder-Decoder 構造を持つ。U-Net は、以下の特徴を持つ。

- Encoder-Decoder 構造 : 特徴マップを Encoder で抽出し、Decoder で元の解像度に戻す
- Skip Connection : 同じ段階の Encoder の特徴マップと Decoder の特徴マップを結合することで、ローカルな情報を復元する
- 損失関数 : クロスエントロピー誤差を用いる

U-Net は、FCN の要素ごとの特徴マップの結合 (Element-wise addition) を行うわけではなく、チャンネル方向で特徴マップを結合する (Concatenation)。このような結合方法を用いることで、FCN とは異なる方法でローカルな情報を復元することができる (図 10)。

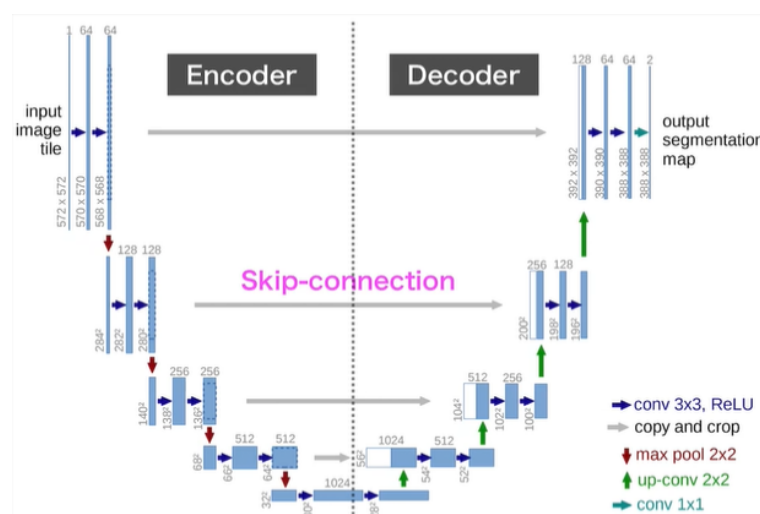


図 10: U-Net のアーキテクチャ

4.5 Dilated Convolution

Dilated Convolution は、畳み込み演算において、フィルタの間に空白を入れることで、Pooling を用いずに受容野を広げる手法である。図 11 に Dilated Convolution の例を示す。

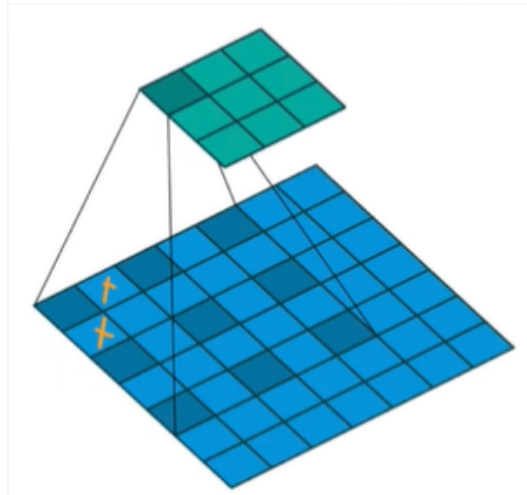


図 11: Dilated Convolution の例: 3×3 カーネルを用いて、 5×5 の畳み込みを行うことで、結果 7×7 の出力を得ることができる。Input の濃くなっている部分がフィルタの適用ピクセルになる。隙間 (rate) を広くすることで、より広い受容野を得ることができる。

5 Instance Segmentation

Instance Segmentation は、Semantic Segmentation の拡張であり、各ピクセルに対して単一のクラスラベルを出力するだけでなく、個体毎にクラスラベルを出力するタスクである。つまり、同じクラスであっても、異なる個体に対して異なるクラスラベルを出力する。

5.1 Mask R-CNN

Mask R-CNN は、物体検知と Instance Segmentation を同時に行うモデルである。物体検出では、物体の同定 (identification: 画像の中で物体がどこにあるのか) と、物体認識/分類 (classification: 物体が何であるのか) を行う。また、Instance Segmentation では、物体の同定と、物体の領域を示すマスクを出力する。この 2 つの作業を同時に行うことで、Mask R-CNN は、画像の中で物体がどこにあり、その物体が何であるのか、そしてさらにその物体の領域を示すマスクを出力することができる。

Mask R-CNN は、R-CNN をベースとしている。R-CNN は、物体検出のみを行うモデルであり、物体の同定と物体認識/分類を行う。R-CNN では、以下の手順で物体検出を行う。

1. 画像を入力する
2. 画像から物体の関心領域 (ROI: Region of Interest) を抽出して、類似する領域をグルーピングする
3. 各 ROI に対して、画像の大きさを揃え、CNN を用いて特徴量を求める
4. 特徴量を用いて、SVM で学習を行い、物体のクラスを推定する

R-CNN は、物体検出の精度は高いが、計算量が多く、推論も遅いという問題があった。この問題を解決するために、Fast R-CNN が提案された。Fast R-CNN では、ROI 毎に CNN に通さず、画像全体を CNN に通す (ROI Pooling) ことで、計算量を削減し、推論速度を向上させた。

さらに、Faster R-CNN が提案された。Faster R-CNN では、物体の関心領域 (ROI) を抽出をも、CNN で行うことで、より高速に物体検出を行うことができるようになった。

Mask R-CNN は、Faster R-CNN にセグメンテーションの機能を追加したモデルである。Mask R-CNN では、画像全体ではなく、物体検出の結果として得られた領域についてのみセグメンテーションを行うことで、Faster R-CNN よりも複雑で細かい領域だけでなく、より高速に出力を得ることができる。

5.1.1 ROI Pooling

ROI Pooling は、Mask R-CNN において用いられた手法で、Fast/Faster R-CNN でも用いられる手法である。ROI Pooling は、畳み込み処理後の特徴マップから、画像全体を固定サイズの特徴マップとして抽出する。

5.2 FCOS

FCOS(Fully Convolutional One-Stage Object Detection) は、物体検出を行うモデルであり、アンカーフリーのモデルである。

アンカーとは Predicted-BBox の位置を決定するための基準となる BBox であり、SSD などを用いられる。アンカーフリーのモデルでは、その基準となるアンカーを用いずに、物体の位置を推定する。具体的には、特

微マップの各ピクセルのうち、Ground Truth の BBox に入っていれば、そのピクセルを物体の中心として、Positive サンプルと判定する。

アンカーを用いる場合、アンカーのサイズ・アスペクト比・数などのハイパーパラメータによって、精度が大きく変動したり、小さな物体に対しての判別が難しくなったり、ネガポジ比率が悪くなるという問題があった。FCOS では、アンカーを用いないことでこれらの問題を解決する。

FCOS は、FPN(Feature Pyramid Network) というアーキテクチャを採用 (図 12) し、異なる解像度の特徴マップを組み合わせることで、様々なサイズの物体を検出することができる。

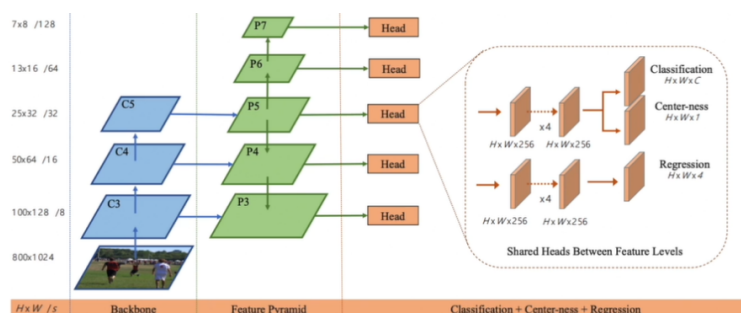


図 12: FCOS のアーキテクチャ

FCOS で学習する内容は

- 物体のクラス (Classification)
- 物体の位置 (Regression)
- 物体の中心 (Centerness)

である。FCOS では、Centerness という概念を導入する。Centerness は、以下の式で定義される。

$$\text{Centerness} = \sqrt{\frac{\min(l, r)}{\max(l, r)} \times \frac{\min(t, b)}{\max(t, b)}} \quad (7)$$

ここで、 l, r, t, b は、Predicted-BBox の中心から左端、右端、上端、下端までの長さを表す。Centerness は中心から離れた位置に、低品質の Predicted-BBox が多数生成される課題にたいして、物体中心に近い位置に高品質の Predicted-BBox を生成するための指標である。

■参考文献

1. 岡谷貴之/深層学習 改訂第2版 [機械学習プロフェッショナルシリーズ]/ 講談社サイエンティフィク/
2022-01-17
2. 1日1分E資格問題 No.23「FCOS」<https://e-shikaku-doujou.com/fcos>