物体検出に用いられるニューラルネットワークモデル
最新モデルのサーベイと目的に応じたモデルの選択
Neural Network Models for Object Detection
A Survey of the Latest Models and Optimal Model Selections for Specific Tasks

金子 純也　　山田 貢己
Junya Kaneko　　Miki Yamada
Morning Project Samurai 株式会社

2021 年 3 月 31 日

keywords:　survey, neural network, object detection, instance segmentation, deep learning

概要

This is English abstract.

1　まえがき

この論文の狙い　物体検出 (object detection) に関する情報を簡潔に纏めたものである．物体検出でできること，必要になるもの，最新の物体検出モデルについて説明した．本論文で引用したライブラリ及び最新情報等は，(★ GitHub アドレスを記載) にある．

物体検出についての本格的な最近のサーベイは文献 [14] に詳しい．

世の中の状況　今や，日常的に AI に囲まれて生活していると言っても過言ではなく，ネットの画像は人工的に加工された画像で溢れ，スマホに話しかけるとあらゆる情報を教えてくれるばかりでなく，電化製品を操作することもできる．自動運転車も増えてきている．

AI 分野における物体検出　ここ数年で飛躍的に性能を向上させた AI 関連技術は，画像認識，物体検出，ロボット制御，音声認識，機械翻訳，ビッグデータ分析などであり，これらの多くの領域で深層ニューラルネットワーク (Deep Neural Network(DNN)) が使われている．とりわけ画像認識分野でこの DNN が注目されるようになったのは，2012 年に開催された最先端の一般物体認識の性能を競うコンテスト ILSVRC において DNN を使った手法が他の手法に大差をつけて優勝したことが発端である．「画像中の物は何か?」に答える物体認識をさらに発展させて，「画像中のどこに何があるか?」に答えようとするものが本論文のテーマ「物体検出」であり，現在の AI ブームを巻き起こした源流がここにあると言ってよい．

物体検出でできること　物体検出とは，カメラで撮影された画像データを電子的に処理し，予め登録しておいた物体 (例えば，人，猫，自動車，...) を見つけ出し，その正確な画像上の位置と物体の種類を予測するものである (「予測 (predict)」は，「推定 (estimate)」，「推論 (inference)」などとも呼ばれ全て同じ意味で使われる)．

現在の標準的方法においては，予め，検出したい対象の学習データ (物体が写っている画像，物体の種類，物体の位置を示す矩形の座標) を大量に用意し，画像を入力すれば種類と位置を出力するように，ニューラルネットワーク等の予測モデルを学習させる．通常，これに多くの時間を必要とする．

学習が完了した予測モデルの能力は，タスクの種類によっては人間の能力 (予測結果のスコアの平均値) を超えたと言われているものもある (物体認識など)．ただし，物体が写し出された画像の品質 (解像度，ノイズ，露出不足/過多) や撮影アングル (遮蔽物，変形，大き (小さ) 過ぎる) に問題がある場合は性能が低下することは避けられない．

物体検出と似た技術として，次の 3 つがある (表 1):

- Semantic segmentation(全画素の物体の種類を認識するが，同種の物体同士は区別しない)
- Instance segmentation(同種の異なる個体を区別して物体検出を行い，且つ，画素単位で個体の識別をする)
- Panoptic segmentation(全画素の物体の種類を認識し，同種の異なる個体も区別する)

表 1　物体検出に関連するニューラルネットワークモデルの分類.

		物体検出	
		する	しない
画素毎の	全画素	Panoptic segmentation [16]	Semantic segmentation
クラス分類	物体領域のみ	Instance segmentation (YOLACT++ [4]，MS R-CNN [12] など)	—
	しない	物体検出 (YOLOv4 [1], EfficientDet [40] など)	—

本論文では上記のセグメンテーション技術も含めた広い意味での物体検出について述べる．

物体検出をするために必要なもの：

ソフトウェア

- PyTorch/TensorFlow 等の深層学習ライブラリとその稼働環境 (Linux/Windows/Mac 上の Python, jupyter notebook 環境など).
- 物体検出を行うソフトウェア (予測モデルの作者，または，物体検出を行おうとする担当者が作ったもの).

ハードウェア

- 前記ソフトウェアが実行できる環境 (PC(GPU があると良い), 或いは，Google Corabolatory などのサーバ上の実行環境).

データ

- 学習データ (事前学習用，及び，fine-tuning 用の入力と出力のペア)
- 本来処理したいデータ (入力).

これらは，使用するソフトウェアで読み取ることのできる状態にしておく (データの前処理).

統計学の知識　物体検出のソフトウェアを使うには，入出力データの意味を理解する必要がある．特に，予測モデルの出力データは通常は誤差を含むものとなるため，出力が表す数値が確率値を表すのか，何らかの物理量を表すのか，分類のカテゴリを表すのか，正確に把握する必要がある．また，学習時の損失関数の値から，予測モデルの推定誤差を見積もることができるのだが，それには出力結果を正しく解釈できる統計学の知識が必要となる．

目的　物体検出は標準的な統計解析の手法よりも手間と計算コストが大きい．もともとしたかったことを明確化して，物体検出の必要性を確認しておく．

2　目的に応じたモデルの選択

2.1　物体検出 (Object detection) と Segmentation(表 1)

物体検出のみを行わせる場合は, YOLOv4(高速で軽量) [1], EfficientDet(検出精度が高い) [40] 等の検出器 (detector) を使用する．

Instance segmentation を行わせる場合は，YOLACT++ (高速) [4]，MS R-CNN(検出精度が高い) [12] 等のモデルを使用する．

Panoptic segmentation を行わせる場合は，Panoptic segmentation 用のモデル [16] を使用する．これは，物体検出器 (object detector) と semantic segmentation モデルの両方を独立に作用させても処理可能ではあるが，一つのモデルとすることにより，処理量を削減できるだけでなく情報量の増加や共通の特徴量の影響の相乗効果で検出性能向上の可能性がある．

2.2　物体検出のオプション技術 (表 2)

表 2 は推論モデル構築の状況に応じて必要となるオプション技術である．

学習データが少ない場合 (数個～十数個) には，few-shot 物体検出が必要になる [15, 6, 42].

モデルをなるべく小さくする必要がある場合は，蒸留をすることでネットワーク規模を削減できる [18].

画像中の重要領域の可視化を行う場合は，可視化の為のニューラルネットワークモデル [34, 35] を適用する．

3　物体検出 (object detection)

3.1　物体検出器 (object detector) の働き

基本動作　物体検出器は，画像 (1 枚の静止画をファイルにしたもの) を処理し，処理結果 (検出個数，検出物体の画像座標，検出物体の種類) を出力する．セグメンテーションの場合は，出力解像度 (画像のサイズ) に応じた各画素のクラス分類結果も出力される．画像を読み込ませる際

表 2　物体検出のオプション技術.

モデル名称	文献	用途	概要	特徴
Feature Reweighting (o) *2	[15]	Few-shot 物体検出	• Few-shot 物体検出器の先駆け. • メタ特徴量学習器，reweighting モジュール，予測モジュールで構成．学習後は reweighting モジュールは削除される.	
Attention-RPN (t) *1	[6]	Few-shot 物体検出	• Attention-RPN, Multi-Relation Detector, Contrastive Training から成る. • few-shot 検出用の 1000 カテゴリのデータセットを作成.	• 再学習と fine-tuning が不要な few-shot 検出.
Two-stage Fine-tuning Approach (TFA) (t)	[42]	Few-shot 物体検出	• Faster R-CNN を用い，最終層のみ fine-tuning する. • Backbone(ResNet, VGG16 等)，RPN，FC sub-network から成る (図 2).	• メタ学習の従来手法よりも 2〜20 ポイント優れている.
Feature mimic method	[18]	物体検出の蒸留	• 物体検出に対して有効な蒸留方法. • 大きなネットワークの特徴マップを教師として学習．大小ネットそれぞれ RoI からサンプリングした特徴を用い，小ネットの特徴は変換層を用いて大ネットのサイズにマッピングする.	物体検出に対して，初めて有効な蒸留の方法を示した.
Grad-CAM	[34, 35]	画像中の重要領域の可視化	• Gradient-weighted Class Activation Mapping (Grad-CAM) を提案．任意の目的概念 (イヌ，人，花瓶，…) の流れの最終 convolution 層での勾配を使い，その概念を予測するため重要な画像中の領域を示す局所化マップを生成. • どのモデルに対しても，アーキテクチャ変更や再学習無しに適用可能.	• Non-attention ベースのモデルであっても，入力画像における判別領域を可視化できる.

*1 (t) : two-stage 検出器　*2 (o) : one-stage 検出器

には，幅と高さを含む情報も与える必要がある．検出器によっては画像サイズや画像ファイル形式が指定されているものもあるのでその場合は，予め画像ファイルを変換する前処理が必要となる.

出力結果の見かた　検出座標は物体を囲む矩形 (bounding box (以下，bbox)) の座標を 4 つの数値で表すことが多い．また，検出の信頼度が 0.0〜1.0 の実数で出力される場合はそれが推定正答確率を表すように設定されている.

学習のしかた　学習データは，予測に必要な入力データに正解出力データを付加したものであり，いわゆる教師あり学習を行わせる．ただ，予測時には無い学習に関するパラメタの設定をしなければならない．検出器の構成 (中間層の層数，特徴量次元のサイズ，検出器独自のパラメタなど) もこの段階で設定する．通常は，確率的降下法でモデルのパラメタを学習させることが多く，検出器の重みパラメタの初期化方法 (平均，分散，値を指定など)，最適化方法 (SGD, Adam, 他)，学習率のスケジューリング，学習打ち切り基準，ミニバッチサイズなどを指定する．学習時には交差検証を行わせて未学習データに対する誤差も計算させることができるので，その誤差を控えておくことにより推論時の予測精度を見積もることができる.

事前学習 (pre-training) と事後学習 (post-training, fine-tuning)　世界最高性能を出すほどの大規模な検出器の学習は，たいてい事前学習と事後学習の 2 段階で行われる．事前学習は主に公開データベースなどの大量データを用いて検出器の前半部分を学習させて適切な内部表現を獲得するために行われる．検出器によっては事前学習済みの重み係数パラメタが公開されているものもある．事後学習は，最終目的に合致したデータを追加して，場合によっては検出器の最終層を追加して，所望の検出処理を実行できるように最終調整の意味合いで実施するものである．事前/事後学習のやり方は各検出器によって異なるため，説明書きや論文等に記された方法を参考にして実行する必要がある.

物体検出器は次の 2 種類に分類できる [14] :

Two-stage 検出器　高い位置決め精度と物体認識精度をもつ (Faster R-CNN など)．最初の CNN ベースの物体検出器 R-CNN とその改良手法はこの方式である．次の 2 つの stage で処理される :

- 第 1 stage: 物体の bbox の候補を選出する.

表 3　物体検出のための主なニューラルネットワークモデル（１）(Segmentation なし).

モデル名称	文献	用途	概要	特徴
ResNet	[11]	DNN	• フィードフォワード型のニューラルネットで最もよく使われる代表的なもの. • 深い階層にもかかわらず高速に学習が可能.	
Feature Pyramid Network (FPN)	[19]	物体検出	• 複数サイズの検出モデルの多くで採用. • コストのわずかな増加で CNN に特徴ピラミッドを導入. • Faster R-CNN と組み合わせて，COCO2016 トップの成績を達成.	物体検出の多くのモデルにおいて，その構造の一部として使われている.
Faster R-CNN (t)*1	[32]	物体検出	• Two-stage 検出モデルの基準となっている．Fast R-CNN を大きく改良したもの. • 画像ピラミッドの代わりに複数サイズの anchor box を用いて複数サイズの物体を検出. • まず RPN だけを End-to-End で学習．次に固定 Anchor と GrandTruth に基づき，物体/背景 (2k)，ずれ (4k) を推論するように全体を学習.	ILSVRC & COCO 2015 Competitions において (segmentation を含む)5 部門で 1 位．さらに，当時としては処理も高速.
RetinaNet (o)*2	[20]	物体検出	• YOLACT においてベースのモデルとして使われている. • one-stage の検出器の新しい損失関数 Focal loss を提案. Focal Loss を導入したシンプルな検出器を RetinaNet と呼ぶ. • Focal Loss は膨大な数の easy negatives が誤差関数に影響を与えすぎることを防ぐ. • ROI の形にかかわらず各レベル同じ領域の情報から推論する．候補が絞られないから Negative 候補の個数は莫大 (10^4〜10^5) になる (one-stage detector の動作の特徴). • 従来の cross entropy は，誤差関数として考えると p>0.6 ならば既に学習済みと考えられるが，曲線はほぼ線形で，p>0.6 の領域でもさらに学習を進めてしまい悪影響を与えていた.	• one-stage detector であるにもかかわらず，性能が従来の two-stage detector の最高性能に追いついた.
YOLOv3 (o)	[31]	物体検出	• YOLOv2 の改良 (one-stage detector). • 320 × 320 YOLOv3 は 22 ms で走り，SSD[23] と同じ精度 28.2 mAP であり，スピードは 3 倍速い. • 過去の YOLO は小さな物体が苦手だったが，YOLOv3 は大きい物体が苦手. • FPN のように 3 つの異なるスケールの特徴量を抽出.	
EfficientDet (o)	[40]	物体検出	• -	• MS COCO Object Detection の AP の最高性能を達成 (YOLOv4 発表時点).
YOLOv4 (o)	[1]	物体検出	• YOLOv3 の改良 (one-stage 検出器). • 以下の新しい特徴を組み合わせて、state-of-the-art の結果を得た： WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, CIoU loss. • 43.5%AP (65.7% AP 50) for the MS COCO dataset at a real-time speed of âĹij65 FPS on Tesla V100.	• EfficientDet 相当の性能を達成しつつ，速度 2 倍を達成. • YOLOv3 と比べて AP を 10 ポイント，FPS を 12 ポイント向上させた.

*1 (t)：two-stage 検出器　*2 (o)：one-stage 検出器

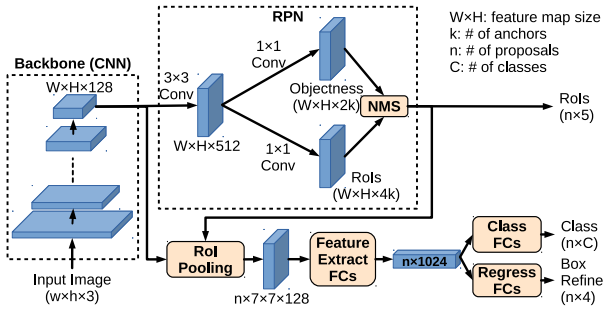

図 1　Faster R-CNN の構造．青い直方体は中間データを表す．矢印と，黄色の枠は処理を表す．

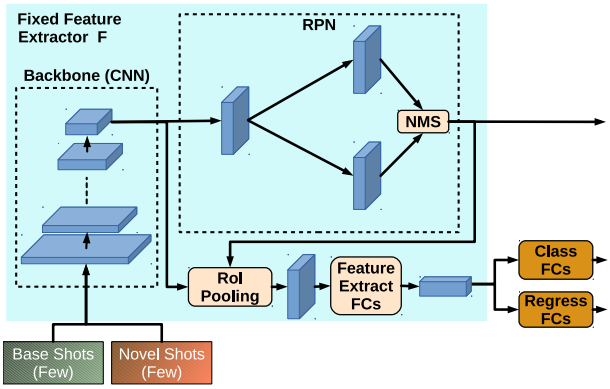

図 2　TFA の Few-shot fine-tuning (2nd stage).

Faster R-CNN では Region Proposal Network (RPN) と呼ばれるものが担当する．

- 第 2 stage: RoI Pooling が各候補 bbox から特徴量を切り出し，後続の分類器 (classifier) と bbox 回帰器 (regressor) がそれを処理する．

One-stage 検出器　推論速度が高速である (YOLO[31, 1], SSD[23] など)．Region proposal をせずに直接，入力画像から bbox を検出して出力する．近年では，改良により検出精度も最高性能を達成している．

3.2　Two-stage 検出器

3.2.1　Faster R-CNN [32]

代表的な two-stage 検出器であり，物体検出に対して当時の最高性能を達成しながら，処理速度の点でも大きく改善したモデルである．PASCAL VOC 2007 に対して mAP=69.9% の精度を 5fps で達成した．

Faster R-CNN の構造　図 1 に示すように，backbone, Region Proposal Network (RPN)，RoI Pooling，特徴抽出ネットワーク，分類器，回帰器で構成される．

Backbone(画像の特徴を抽出する処理 (縦横サイズを小さくしながら特徴量を含むチャネルを増やしていくこと)) は CNN で構成され，$W \times H \times 128$ のサイズの特徴量を出力する．提案当時は VGG-16 [36] などが用いられた．

RPN は backbone が出力した特徴量をさらに CNN で処理して物体の種類に依存せずに物体を検出してその候補 (proposal と呼ぶ) の RoI (bbox 座標) と物体らしさ (objectness) を n 個ずつ出力する．RPN 内部では特徴量を CNN で変換し，特徴量の各空間座標点毎に k 個の「bbox(4 個) と物体らしさ (2 個) の数値」を同時に出力している (計 $WH(4+2)k$ 個)．これらの bbox は non-maximum suppression (NMS) によって，物体らしさが閾値を超えているものだけを残し，また，1 個の物体を複数回検出したものも 1 個だけに集約することにより，RPN が出力する proposal を n 個以下に抑える．ここで，各座標点に割り当てられた k 個の初期 bbox を anchor box と呼んでいる．

RoI Pooling は，backbone からの特徴量と RPN からの n 個の proposals を受け取り，bbox 内部の特徴量を切り出して物体の大きさにかかわらずに空間サイズを 7×7 の固定サイズに変換する．

固定サイズの特徴量は，共通の特徴抽出を行う全結合 (Full Connect) 層 (以下，FC 層) により $n \times 1024$ のデータに変換され，分類器と回帰器に送られて別々の FC 層で処理されて，クラス分類確率と bbox 調整量がそれぞれ出力される．(★ bbox 調整量が全クラス分出力されるのかどうか確認要！)

Faster R-CNN の学習

1. CNN による特徴量生成は，画像認識などの学習で構築された重みを初期値として fine tuning を行う．
2. RPN だけを End-to-End で学習．anchor と教師データに基づき，物体か背景か (物体らしさ，2k 個) と，anchor からのずれ (4k 個) を学習．
3. RPN 固定で全体を学習する．RPN の RoI 出力を全体の出力及び RoI pooling へ送り，特徴抽出器で分類と回帰の共通部分の FC 計算を行う．分類器はクロスエントロピー，softmax などで計算し，回帰器が行う線形回帰は L1 ノルムで学習し，bbox の位置の補正を行う．

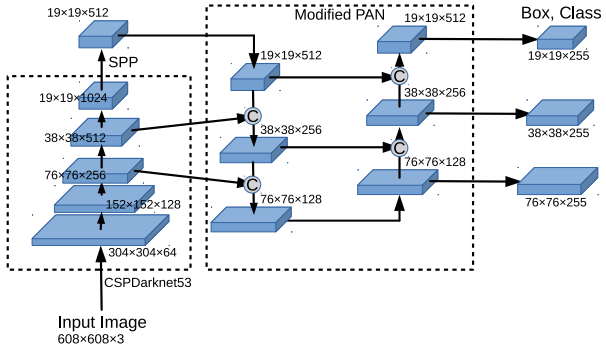

図 3　YOLOv4 の構造．青い直方体は中間データを表し，矢印は処理を表す．

3.3　One-stage 検出器

3.3.1　YOLOv4 [1]

代表的な one-stage 検出器であり，高速，高性能で軽量であることが特長である．YOLOv3[31] までの特徴を継承しつつ，採用可能な backbone，neck, head から最適な組み合わせを選択し，また，適用可能な各種技術を取り込んで構成された (図 3).

YOLOv4 の構造　入力画像は，backbone，neck(PAN や BiFPN のように、特徴ピラミッドを昇り降り横断することで特徴量を統合する)，head(「backbone + neck」が出力する特徴量から、クラス信頼度と bounding box オフセットを予測する) の順に処理される．

backbone である CSPDarknet53 は，YOLOv3 で使われた Darknet53 に Cross Stage Partial Network (CSP) を導入したものである．Darknet53 は，residual 結合を持ち Convolution 2 層で構成されるブロックが多数積み重なった構造であり，名前の 53 は Convolution が 53 個あることから来ているとしている (ただし，53 番目の層は全結合層である)．CSP network は予測性能を向上させる工夫であり，注目するブロックへの特徴量入力を 2 つに分割し，一方はそのブロックで処理し，もう一方は処理をスキップしてそのまま送り，これら 2 つを連結 (concatenation) して出力することを行う．ただし，YOLOv4 においては，この「分割」処理の代わりに，分岐させた直後に stride=2 の Convolution で処理して両方のチャネルサイズを 1/2 にすることで適用している．

CSPDarknet53 の Top の出力は SPP モジュール (kernel size=1, 5, 9, 13, stride=1 として，4 つ並列に max pooling を行い，これらを concat するもの) に送られる．比較的大きな k×k max-pooling が効果的に backbone 特徴量の受容野を増加させてから，neck の Top に渡される．

neck である Modified PAN は，Path Aggregation Network (PAN)(3 つの特徴ピラミッドを行き来して高解像度情報を特徴マップに効果的に伝えるモデル) における bottom-up path の加算計算を concatenation に変更したモデルが用いられている．

ここではさらに，Modified Spatial Attention Module (SAM)(Convolution の出力を 2 つに分岐し，一方に Convolution + sigmoid 処理を行い，元信号に掛け算する) の演算を行ったものが PAN の 3 個の出力として head に渡される．

3 つの head は，それぞれ独立に convolution 計算を 2 回行い，最終的なクラス信頼度と bbox オフセットを出力する．

YOLOv4 の学習　一般的な確率的降下法を用いた学習に加えて，いくつか効果的な学習の工夫を導入している:

CutMix　学習画像の一部を切り取ったパッチを別の学習画像の一部に貼り付けて，正解ラベルもパッチの面積に比例させてミックスしたものを生成してそれで学習する．

Mosaic データ拡張　CutMix を，4 つの画像を用いるように拡張したもの．

DropBlock　特徴マップに対して，無作為に選出した矩形範囲 (block) にマスクを掛けて無効化したもので学習する．

Class label smoothing　クロスエントロピー損失で学習する際に，logit が発散しないように正解ラベルを $q'(k|x) = (1-\epsilon)\delta_{k,y} + \epsilon/K$ とした損失を用いて学習する．

Complete IoU (CIoU) 損失　二つの bbox が離れていても，近づきすぎていても適切な損失関数になるように，IoU 損失を改良した損失関数．

CmBN　batch normalization を (ミニバッチ複数回につき 1 回の重み更新を行う場合に) ミニバッチをまたいで「平均，分散」を蓄積して，重み更新と同じタイミングで bias と scale を更新する．

Self-adversarial-training (SAT)　まず，物体を写した元画像を改変し，(物体であるにもかかわらず) 物体が存在しない画像であると騙される画像を生成する．次に，この改変された画像中の物体を検出するように通常の学習を行う．

Cosine annealing scheduler　コサイン関数の形状を利用して，学習率を少しずつ減少させる．これを周期を増

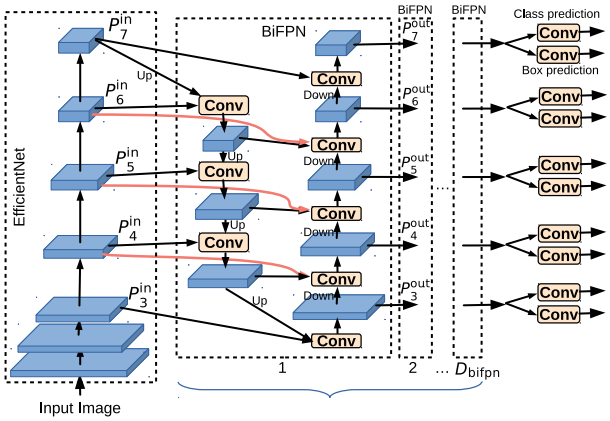

図 4　EfficientDet の構造．(Conv) は convolution を含む複合処理を表す．BiFPN は D_{bifpn} 段だけ反復する．赤い線の接続はスキップ接続 (weighted residual connection (WRC)) を示す．Class/box 予測は各レベル毎に実行されるが，重みは共通．

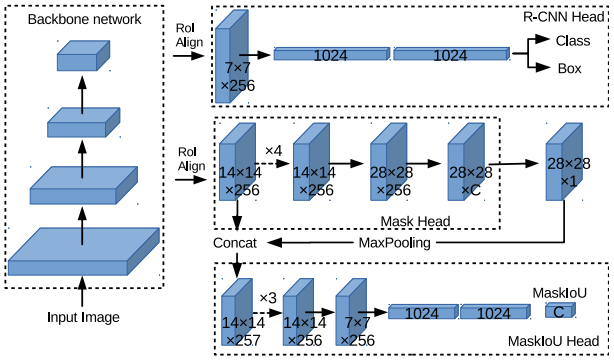

図 5　Mask Scoring R-CNN の構造．

やしながら複数回繰り返す．

3.3.2　EfficientDet [40]

2 つの手法 (マルチスケールの特徴量の融合を行わせる weighted bi-directional feature pyramid network (BiFPN) と，物体検出器の構造設計選択を系統的に行い効率性を向上させる最適化方法 (compound scaling 法)) を提案し，これらの提案手法と EfficientNet backbone [39] に基づき，物体検出器の新しい系列 (EfficientDet) を開発した．

EfficientDet D7 は，COCO test-dev に対して 52.2 AP という最高性能を，52M パラメタ，325B FLOPs で達成 (同じ AP での比較では，パラメタ数は従来よりも 1/4〜1/9 だけ小さく，処理時間は 1/13〜1/42 だけ少なくなる)．

EfficientDet の構造　backbone として画像分類モデルの EfficientNet[39] を用い，中間の特徴量の統合を BiFPN を多段に接続させたもので実行し，クラスと bbox の予測は RetinaNet[20] と同様の構造 (全レベルで重みを共有) を採用した．

BiFPN は $D_{\text{bifpn}}(=3,\cdots,8)$ 段に接続され，Up と Down の双方向の接続が何度も反復される．特徴量の統合は次式による fast normalized fusion と呼ぶ方法が使われている：

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i$$

「backbone，BiFPN，box/class 予測ネットワーク」の，解像度，深さ，幅 (チャネル数) 等のモデルの規模を決めるパラメタを，1 個の compound 係数 ϕ を用いて連動して変更する．メモリ量と FLOPs の制約内で，accuracy を最大化するパラメタ数の配分を求め，それを保ちながらモデル規模を変更することができる．

EfficientDet の学習　SGD を momentum=0.9，weight decay=$4e^{-5}$ で実行．学習率は第 1 エポックは 0 から 0.16 まで線形に増加させ，その後 cosine で減衰させるアニーリングを行う．活性化関数は swish，損失関数は focal loss を使用．

4　インスタンスセグメンテーション (Instance segmentation)

4.1　Mask Scoring R-CNN (MS R-CNN) [12]

マスク品質 (インスタンスマスクと正解マスクとの IoU として定量化されるもの) を分類スコアと明示的に関連付けたモデルである．MS R-CNN は，予測マスクの品質を学習するためのブロック (MaskIoU head) を，Mask R-CNN[8] に導入したモデルになっている (図 5)．MaskIoU head はインスタンスの特徴量と対応する予測マスクを一緒に取り込み，それを元に Mask IoU を回帰推定する．そして，推論時にこの予測 MaskIoU を分類スコアに掛け算して補正する．

4.1.1　MS R-CNN の学習

MaskIoU の学習サンプルとして RPN proposals を使う．proposal box と正解 box との IoU が 0.5 以上の学習サンプルが必要となる．これは Mask R-CNN の Mask head の学習サンプルの場合と同じである．各学習サンプルに対する回帰目標を生成するために，まず目標クラスの予測マスクを取得し，予測マスクを閾値=0.5 で 2 値化す

表 4　物体検出のための主なニューラルネットワークモデル（２）(Segmentation あり).

モデル名称	文献	用途	概要	特徴
Mask R-CNN (t)*1	[8]	Instance Segmentation	• Faster R-CNN を拡張し，並列にマスクを予測するブランチを追加．PANet のベースとなるモデル． • Backbone は FPN が使われることがある． • 計算量は Faster R-CNN より少し増える程度．	• Bbox 検出も併用していることで，安定した性能を達成．
Path Aggregation Network (PANet) (t)	[22]	Instance Segmentation	• Mask R-CNN + FPN に改良を加えた． • Bottom-up path augmentation により下位層から上位層への情報経路を短くして，元画像の正確な位置情報を特徴量と関係づける． • Adaptive feature pooling が全ての特徴レベルをリンクして直接後続に伝える．	• 多くの改良点の合わせ技で数ポイントの精度向上を達成． • COCO2017 Instance Segmentation で 1 位相当の性能を達成．
Mask Scoring R-CNN (MS R-CNN) (t)	[12]	Instance Segmentation	• 予測マスクの品質を学習する MaskIoU Head を備えた，Mask Scoring R-CNN (two-stage 検出器) を提案． • MaskIoU Head によりマスク品質を回帰推定する． • 推論時に、推定マスク品質を分類スコアに掛け算して補正し，マスク品質が悪いのに分類スコアが大きくなってしまうことを抑制する．	• Mask R-CNN を抜いて，インスタンスセグメンテーションの最高性能を達成．
YOLACT++ (o)*2	[4]	Instance Segmentation	• RetinaNet (物体検出) にブランチを幾つか追加してマスク予測機能を持たせたモデル (one-stage 方式)． • 実時間 (>30 fps) Instance Segmentation であって，MS COCO に対して最高性能相当を達成． • Fast NMS を提案し，12ms ほど計算時間を短縮した． • Deformable convolution を backbone に導入． • Re-Scoring Network を導入してマスク品質に基づいてマスク予測を再格付けする等の改良． • 34.1 mAP on MS COCO at 33.5fps を達成． • 検出 anchor の (スケールとアスペクト比の) 選択を工夫．	• 実時間動作を満たしつつ，Mask R-CNN に近いセグメンテーション性能を達成．
Panoptic Segmentation	[16]	Panoptic Segmentation	• 画像の全画素に対してクラス分類を行い (semantic segmentation)，且つ，物体に関しては個体を区別して画素単位で識別する (instance segmentation)． • 単に，semantic segmentation と instance segmentation の両方を行えば済むかもしれないが，学習時に両方の教師データが与えられることは片方だけの場合に比べて情報量が多いため，一つのモデルで両方の処理を行わせた場合に推論精度が良くなる可能性がある (★裏を取る必要あり)．	• -

*1 (t) : two-stage 検出器　*2 (o) : one-stage 検出器

る．そして，2 値化マスクと正解との MaskIoU を使う．MaskIoU を回帰するのには L2 損失を使い，損失重みは 1 にする．ネットワーク全体は end-to-end で学習する．

4.1.2　MS R-CNN の推論処理

MaskIoU Head は分類スコア (R-CNN head の出力) の調整に使う．推論の手順は次のようになる:

1. R-CNN head が N 個の bbox を出力する．
2. N 個の bbox のうち，SoftNMS[2] で上位 k 個の bbox を選択する．
3. 上位 k 個の bbox を Mask Head に入力し，k 個のマルチクラスマスクを生成する (ここまでは標準的 Mask R-CNN の手順)．
4. これら k 個のマスクを目標として MaskIoU Head に入力し，予測 MaskIoU を出力する．
5. 予測 MaskIoU を，分類スコアに掛け算し，上位 k 個の修正された分類スコアを得る．

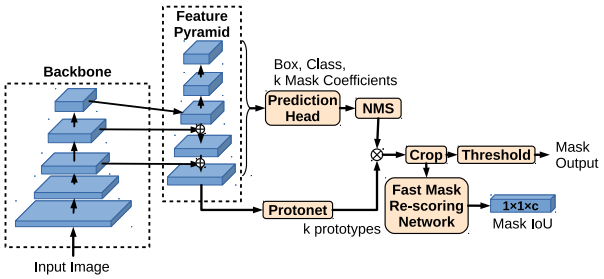

図 6　YOLACT++ の構造．青い直方体はデータを表す．矢印と，黄色の枠は処理を表す．

4.2　YOLACT++ [4]

実時間 (>30fps) で動作するインスタンスセグメンテーションのモデルであり，MS COCO に対して当時の最高性能に匹敵する性能 (34.1 mAP at 33.5fps) を達成した．fully-convolution モデルであり，deformable convolution を YOLACT[3] の backbone に導入する等の改良をしている。物体検出モデルの RetinaNet[20] をもとに，インスタンスセグメンテーション向けに改良したものである (図 6)．

4.2.1　YOLACT++ の学習

Easy negative が多くて学習が困難になる問題は，OHEM 法[3]を用いて negative : positive = 3:1 にして学習することで対応する．Class 信頼度は分類損失 (クロスエントロピー)，bbox は L1 損失，mask (「mask 係数×Prototype」で得られるもの) は pixel-wise binary cross entropy でそれぞれ学習する．Re-Scoring Net は Mask IoU(係数) を回帰する学習を行う．

Semantic segmentation 損失は，学習時のみ接続されるネットワークの学習であり，この学習の実施により mAP が 0.4 ポイント向上する．P3 特徴量出力に 1×1 convolution 1 層で処理して c チャネルの出力をさせて最後にシグモイド関数をかける．これが正解 mask になるように学習する．

4.2.2　YOLACT++ の推論処理

Prediction head が各 anchor のクラス信頼度，bbox，k 個の mask 係数を出力し，Protonet が k 個の Prototype (mask) を出力する．そして，Predition head 出力を NMS 処理して選ばれた結果に対して，mask 係数と Prototype を積和して全画面の mask を得る．この mask において予測 bbox の外側を 0 で埋めたものを 2 値化することで，最終的な予測 mask 出力を得る．

並行して，2 値化する前の mask を Re-Scoring Net に入力して，mask IoU 出力を得る．分類スコアは，この mask IoU を掛けて補正される．

[3] 入力画像に対する全 RoI をミニバッチと考えて、損失の値でソートして識別が難しい negative を選択して学習させる方法．

5　パノプティックセグメンテーション (Panoptic segmentation)

[16].

6　物体検出に使われる性能向上技術

表 5 と表 6 に，物体検出のモデルと組み合わせて用いられる主な性能向上技術を示した．同じ物体検出モデルであってもそのモデルを構成する各種モジュールや，そのモデルと組み合わせる各種性能向上技術の利用有無によって予測性能は変化する．性能を最大化する各種技術の組み合わせにより 10 ポイント程度の AP の上積みができる場合があり，最適な組み合わせはタスクやデータの種類によって変化するため，文献に記載された期待性能が出ない場合などには，そのタスクに最適な組み合わせをあらためて探索する必要がある．

6.1　部分構造，特徴量生成

Cross stage partial network (CSPNet) [41]　ある経路を流れる特徴量を 2 分割し，片方だけ次の処理ブロックで処理した後に，処理していないもう片方の特徴量と連結する．性能をほとんど低下させずに処理を軽くすることができる．ResNet, ResNeXt, DenseNet に基づくアーキテクチャを扱える汎用性がある．ImageNet データセットでは同等以上の精度で計算量を 20% 削減．MS COCO データセットでは AP50 の場合に最高性能を達成．

Spatial pyramid pooling (SPP) [10, 9]　Pooling 戦略の一つであり，画像のサイズによらずに複数サイズの受容野に対応した複数の特徴量 (固定長) を生成する．Pascal VOC 2007 に対して，当時の最高検出性能を，R-CNN よりも 24〜102 倍速い処理速度で達成．

Atrous spatial pyramid pooling (ASPP) [5]　特徴抽出において解像度が縮小されないように，upsampling フィルタを用いた畳み込み (atrous convolution / dilated convolution) と SPP を組み合わせた構造である．PASCAL VOC-2012 semantic image segmentation task にて 2016 年当時の最高性能を達成．

6.2　特徴量の統合

Multi-input weighted residual connections (MiWRC)[40]　PAN 構造において，backbone 側の各レベルの特徴量からトップダウン経路をスキップして出力側のもう一本の

表 5　物体検出の性能向上技術（1）.

技術名称	文献	用途	概要	特徴
Cross Stage Partial Network (CSPNet)	[41]	予測性能向上 (及び CNN の軽量化)	• 分割した特徴量の片方をある処理ブロックで処理し，未処理の特徴量と連結．性能を低下させずに処理を削減. • ResNet, ResNeXt, DenseNet を扱える汎用性を持つ.	• ImageNet データセットでは同精度で計算量を 20% 削減.
Spatial Pyramid Pooling (SPP)	[10, 9]	予測性能の向上 (特徴量生成)	• Pooling 戦略の一つであり，画像のサイズによらずに複数サイズの受容野に対応した複数の特徴量 (固定長) を生成する.	• PascalVOC2007 の最高性能．R-CNN の 24 倍超高速.
Atrous spatial pyramid pooling (ASPP)	[5]	予測性能の向上 (特徴量生成)	• 特徴抽出において upsampling フィルタを用いた畳み込みと SPP を組み合わせた.	• PascalVOC-2012 で 2016 年当時の最高性能.
Multi-input weighted residual connections (MiWRC)	[40]	予測性能の向上 (特徴量の統合)	• PAN 構造において，backbone 側からトップダウン経路をスキップして，出力側のもう一つのボトムアップ経路の同じレベルへのスキップ接続を設けること.	
Scale-wise Feature Aggregation Module (SFAM)	[47]	予測性能の向上 (特徴量の統合)	• 全特徴ピラミッドから特定スケールの特徴マップを取り出して連結し，Global average pooling で 1×1×1024 の重みを作成して特徴量を修正 (attention 処理).	
Adaptively Spatial Feature Fusion (ASFF)	[21]	予測性能の向上 (特徴量の統合)	• point-wise level re-weighting (空間座標毎の重み付け) を用いて異なるスケールの特徴マップを統合する.	
Weighted Bi-directional Feature Pyramid Network (BiFPN)	[40]	予測性能の向上 (特徴量の統合)	• PANet に MiWRC(図 4 の赤の接続) を追加する等の改良を加えた構造．スケール毎の重みで level re-weighting を実行して特徴マップを統合する．BiFPN は多段に接続される.	
Convolutional Block Attention Module (CBAM)	[43]	予測性能の向上 (Attention)	• 任意の CNN に適用可能な 2 種類の attention module. • Channel attention module (CAM) は，空間方向の global pooling で各チャネルの重みを計算して attention 処理. • Spatial attention module (SAM) は，チャネル方向の global pooling で各空間座標の重みを計算して attention 処理.	• 全ての場合において，ベースラインよりも性能を向上させた.
Batch normalization	[13]	学習性能の向上 (バッチ正規化)	• 学習時に各ミニバッチに対する入力分布を正規化することをモデルの一部として組み込んだもの.	• 高い学習率が可能になり，学習を飛躍的に加速させる.
Cross-iteration batch normalization (CBN)	[44]	学習性能の向上 (バッチ正規化)	• 複数の iteration の統計量 (平均，標準偏差) を現在の iteration から線形近似して予測 (重みの更新の単位を iteration としている).	• 統計量の推定精度を向上させる.
Cross mini-batch normalization (CmBN)	[1]	学習性能の向上 (バッチ正規化)	• (ミニバッチ複数回につき 1 回の重み更新を行う場合に) ミニバッチをまたいで「平均，分散」を蓄積して，重みの更新と同じタイミングで，蓄積した「平均，分散」に基づいて bias と scale を更新する.	• 計算コストをほとんど増やさずに，統計量の推定精度向上が図れる.
Soft-NMS (non-maximum suppression)	[2]	予測性能の向上 (NMS)	• 一部重なっている 2 つ以上の bbox の IoU が閾値を超えている場合，bbox 候補を削除する代わりに，そのスコアにペナルティを与える.	• 近接した物体に対する検出性能を向上させる.

表 6　物体検出の性能向上技術（2）.

技術名称	文献	用途	概要	特徴
DIoU-NMS	[48]	予測性能の向上 (NMS)	• Distance-IoU 損失を用いて NMS 処理を行う. • DIoU は 2 つの bbox の中心間距離を考慮し，一部重なっている別の物体を区別しやすくしている.	• 重なっている別の物体に対する検出性能を向上.
Mish activation	[26]	予測性能の向上 (活性化関数)	• 活性化関数 $f(x) = x\tanh(\ln(1+e^x))$ を使う.	
Distance-IoU (DIoU) 損失/Complete IoU (CIoU) 損失	[48]	学習性能の向上 (損失関数)	• 予測 box と目標 box との正規化距離を用いた Distance-IoU (DIoU) 損失と，さらに，アスペクト比の損失を追加した Complete IoU (CIoU) 損失を提案. • DIoU は NMS に簡単に適用できる.	• IoU 損失，GIoU 損失より学習の収束が速く性能が良い.
Label smoothing	[38]	学習性能の向上 (損失関数)	• クロスエントロピー損失で学習する際に，正解ラベルを $q'(k\|x)=(1-\epsilon)\delta_{k,y}+\epsilon/K$ ($\epsilon>0$) とおいて logit が発散して不安定化することを防ぐ.	
Dimension cluster	[30]	学習性能の向上 (パラメタの初期値)	• Anchor box の初期値の設定方法 • 学習セットの bbox に対して k-means クラスタリングを用いて，anchor box の良い事前分布を取得.	• 学習を高速化し検出性能を向上させる.
DropOut	[37]	学習性能の向上 (DropOut)	• 過学習を抑制する. • 学習時に毎回ランダムに α ($0 \leq \alpha < 1$) の割合でノードを無効化し，実行時には出力を α 倍する.	
DropBlock	[7]	学習性能の向上 (DropOut)	• Convolution 層に対して有効な，構造化された dropOut 方法．特徴マップの連続した領域をドロップさせる.	• COCO detection で RetinaNet の AP が 1.6 ポイント向上.
mixup	[46]	学習性能の向上 (データ拡張)	• 2 つの教師データと，配分比率 λ (0.0〜1.0) を用いて，入力画像とラベル (one-hot encoding) の両方を $\tilde{x} = \lambda x + (1-\lambda)x$ で線形結合して新たな教師データを生成する.	
CutMix	[45]	学習性能の向上 (データ拡張)	• 学習画像のパッチを別の学習画像にカット＆ペーストして 1 枚の画像に 2 種類の画像を混在させ，正解ラベルもパッチの面積に比例させてミックスする.	• Pascal VOC に対して mixup,Cutout より高性能.
Mosaic data augmentation	[1]	学習性能の向上 (データ拡張)	• 異なる学習画像 4 枚のパッチをモザイク状に並べて，正解ラベルもパッチの面積に比例させてミックスする (CutMix の 4 枚版).	• YOLOv4 に対しては，CutMix よりも良い性能を示す.
Self-adversarial training (SAT)	[1]	学習性能の向上 (データ拡張)	• 学習が困難になるように学習データの入力を改変. • GAN や adversarial attack とは異なり，通常の検出性能向上が目的.	• YOLOv4 の mAP が 46.7%→50.5% に向上.
SGDR (Cosine annealing scheduler)	[24]	学習性能の向上 (高速化)	• 周期的に SGD の warm restart をする．各 restart にて，学習率を徐々に小さくするようにスケジューリングする.	• 同じ性能に至るまでの epoch 数が 1/2 以下で済む.

ボトムアップ経路の同じレベルへのスキップ接続を設けること (図 4 の赤線で示された接続).

Scale-wise Feature Aggregation Module (SFAM) [47] の処理手順:

1. 異なる特徴抽出レベルの複数の特徴ピラミッド 1, 2, $\cdots, t$ を生成.
2. 全ての特徴ピラミッドから，特定のスケールの特徴マップを取り出して連結したものを作成.
3. 連結した特徴量を，Global average pooling でサイズ 1×1×1024 にして (Squeeze)，さらに 2 つの FC 層 (Excitation) で変換して重みを作成 (重要なチャネルは学習により大きな重みになっているものと考える).
4. 重みを用いて，Squeeze する前の特徴量を修正.

Adaptively Spatial Feature Fusion (ASFF) [21] の処理手順:

1. (Level 1,2,3 から成る) 特徴ピラミッドが得られているものとする.
2. 各レベルの特徴量を統合した Level l の特徴量を $\boldsymbol{y}^l$ とする．$\boldsymbol{y}^l$ を構成する空間座標 (i, j) における特徴量 $\boldsymbol{y}^l_{ij}$ を次式で計算する:

$$\boldsymbol{y}^l_{ij} = \alpha^l_{ij}\boldsymbol{x}^{1\to l}_{ij} + \beta^l_{ij}\boldsymbol{x}^{2\to l}_{ij} + \gamma^l_{ij}\boldsymbol{x}^{3\to l}_{ij}.$$

   ただし，$\boldsymbol{x}^{n\to l}_{ij}$ は Level n から Level l にリサイズされた特徴量，$\alpha^l_{ij}, \beta^l_{ij}, \gamma^l_{ij}$ は全チャネルで共有されるスカラー変数.
3. 生成された $\boldsymbol{y}^l$ を各レベルの後段の検出器に送る.

ASFF は空間座標毎の重み付けをしており，point-wise level re-weighting と呼ばれる.

Weighted Bi-directional Feature Pyramid Network (BiFPN)[40]. PANet に対して，入力から出力 node への接続 (MiWRC(図 4 の赤の接続)) を追加する等の改良を加えた構造．スケール毎に定められた重みで level re-weighting を実行し，異なるスケールの特徴マップの荷重和を計算して特徴マップを統合する．BiFPN は多段に接続され，性能が最大になるように段数を最適化する.

6.3 Attention

Convolutional Block Attention Module (CBAM)[43]. CNN のための，単純で効果的な attention module．任意の CNN に適用可能で，オーバーヘッドは殆ど無い．分類と検出の評価実験において一貫して改良を示している．情報表現の可視化にも利用可能．channel attention module (CAM) と spatial attention module (SAM) を実行して CNN の中間の特徴量を洗練させる.

CAM は，空間方向に global max pooling と global average pooling を行い，それぞれ全結合層を通したものの和に sigmoid 関数を作用させて，各チャネルの重みを求めて，attention 処理をする.

SAM は，チャネル方向に global max pooling と global average pooling を行い，それらを連結したものに convolution を実行してチャネルサイズを 1 にする．さらに sigmoid 関数を作用させて，各空間座標の重みを求め，その重みを用いて attention 処理をする．YOLOv4 においては，2 種類の pooling を 1×1 convolution に置きかえる等の改変を行った (簡素な構成の) modified SAM が用いられている.

6.4 バッチ正規化

Batch normalization[13]. ニューラルネットワークに内在する内部共変量シフトの問題を解消するため，学習時に各ミニバッチに対する入力分布を正規化することをモデルの一部として組み込んだものである．高い学習率を設定することが可能になり，学習を飛躍的に加速させることができる．深層学習の手順の一つとして必須の標準的手法となっている.

Cross-iteration batch normalization (CBN)[44]. 直近の複数の iteration における平均や標準偏差などの統計量と，現在の iteration のそれとの関係を線形近似で求め，それを用いてバッチ正規化で用いる統計量を補正する (重みの更新の単位を iteration とする)．統計量の推定精度の向上が図れる.

Cross mini-batch normalization (CmBN)[1]. ミニバッチ複数回につき 1 回の重み更新を行う場合に，ミニバッチをまたいで「平均，分散」を蓄積して，重みの更新と同じタイミングで，蓄積した「平均，分散」に基づいて bias と scale を更新する．計算コストをほとんど増やさずに，統計量の推定精度向上が図れる.

6.5 Non-maximum suppression (NMS)

NMS は物体検出における必須の手順であり，同一の物体に対して微妙に異なる位置に検出された複数の検出候補を，1 個の検出結果に集約する処理である．(★これの標準的手順ってどうやるんだっけ？物体らしさとクラス確率のどっちを使うんだっけ？文献はどれ？)

Soft-NMS (non-maximum suppression) [2]. 一部重なっている 2 つ以上の bbox の IoU が閾値を超えている場合，bbox 候補を削除する代わりに，そのスコアにペナルティを与える．IoU とスコアの両方を考慮することにより，近接した物体に対する検出性能を向上させる.

DIoU-NMS [48]. Distance-IoU (DIoU) 損失を用いて NMS 処理を行う．DIoU は 2 つの bbox の中心間距離を考慮し，一部重なっている別の物体を区別しやすくすることで検出性能を向上させる．

6.6 活性化関数

通常は，Sigmoid 関数 [33] や ReLU 関数 [27] 等が使われ，DNN における非線形性を担っている．

Mish activation [26]. 活性化関数として

$$f(x) = x \tanh(\ln(1 + e^x))$$

を用いるものであり，$\pm\infty$ において ReLU 関数に漸近する性質がある．使用する識別モデルによって結果は異なるが，CIFAR-10 及び ImageNet-1k に対する画像分類タスクに関して，ReLU や Swish[29] よりも好成績をおさめる場合がある．

6.7 損失関数

Distance-IoU (DIoU) / Complete IoU (CIoU) [48]. 予測 box と目標 box との正規化距離を用いた Distance-IoU (DIoU) 損失を提案（従来は Ln 損失等が使われることも多かった）．DIoU は IoU 損失や GIoU 損失より学習の収束が速い．さらに，追加でアスペクト比を直接損失に組み込んだ，収束が速く性能が良い Complete IoU (CIoU) 損失を提案．DIoU と CIoU を，YOLOv3, SSD[23], Faster R-CNN に導入することで，性能を向上させることができる．DIoU は NMS に簡単に適用することができる．

Label smoothing[38]. クロスエントロピー損失で学習する際に，正解ラベルを $q'(k|x)=(1-\epsilon)\delta_{k,y} + \epsilon/K \quad (\epsilon>0)$ とおいて logit が発散して不安定化することを防ぐ．

6.8 初期値の設定

Dimension cluster [30]. 学習セットの bbox に対して k-means クラスタリングを適用して，自動的に anchor box の良い事前分布を取得し，それに従って anchor box の初期値を設定する (「dimension」は寸法という意味)．学習を高速化するだけでなく，検出性能も向上させる．

6.9 DropOut

DropOut[37] は学習時に毎回ランダムに $\alpha\ (0 \leq \alpha < 1)$ の割合でノードを無効化して学習し，実行時には全ノードを有効化する代わりに各ノードの出力を α 倍することを行うことで過学習を抑制する．DNN の為の強力な正則化法の一つである．

DropBlock [7]. Convolution 層に対して有効な，構造化された DropOut 法．特徴マップの連続した領域をドロップさせる．ImageNet 分類では，ResNet-50 に適用すると 1.6 ポイント精度が向上．COCO detection では RetinaNet の AP を 36.8% から 38.4% に向上させた．

6.10 データ拡張

データ拡張 (data augmentation)[17] とは，学習データを変化させたもの (平行移動，左右反転，色変換など) を多数生成して擬似的に学習データの個数を増やすことによって，過学習を抑制する方法である．

mixup [46]. 2 つの教師データと，配分比率 λ (0.0〜1.0) を用いて，入力画像とラベル (one-hot encoding) の両方を $\tilde{x} = \lambda x + (1 - \lambda)x$ で線形結合して新たな教師データを生成する方法である．

CutMix [45]. 学習画像のパッチを別の学習画像にカット＆ペーストして 1 枚の画像に 2 種類の画像を混在させ，正解ラベルもパッチの面積に比例させてミックスする．ImageNet(Cls), ImageNet(Loc), Pascal VOC Detection に関して，代表的な従来法（Mixup, Cutout）よりも良い性能を示す．

Mosaic data augmentation [1]. 異なる学習画像 4 枚のパッチをモザイク状に並べて，正解ラベルもパッチの面積に比例させてミックスする (CutMix の 4 枚版)．YOLOv4 に対しては，CutMix よりも良い性能を示す．

Self-adversarial training (SAT) [1]. 学習が困難になるように学習データの入力値を改変した学習データを生成し，それを学習することで検出性能を向上させる．Generative Adversarial Networks (GAN) や adversarial attack などとは異なり，あくまでも通常の検出性能向上を目的とする．mAP が 46.7%→50.5% に向上したという実験結果がある．

6.11 学習の高速化

SGDR (Cosine annealing scheduler) [24]. 周期的に SGD (Stochastic gradient descent) の warm restart をし，restart の回を追う毎に学習率を徐々に小さくするようにスケジューリングする．Warm reatart 付きの SGD は，従来法に比べて，同じ性能に至るまでの epoch 数が 1/2 から 1/4 で済む．

7 むすび

謝辞について (?) どうする？

参考文献

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020.

[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS — Improving object detection with one line of code. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5562–5570, 2017.

[3] D. Bolya, C. Zhou, F. Xiao, and Y. Lee. YOLACT: Real-time instance segmentation. In *International Conference on Computer Vision (ICCV 2019), Seoul*, pp. 9156–9165, October 2019.

[4] D. Bolya, C. Zhou, F. Xiao, and Y. Lee. YOLACT++: Better real-time instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, 2020. DOI: 10.1109/TPAMI.2020.3014297.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, pp. 834–848, 2017. DOI: 10.1109/TPAMI.2020.3014297.

[6] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai. Few-shot object detection with attention-RPN and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA*, pp. 4013–4022, June 2020.

[7] G. Ghiasi, T.-Y. Lin, and Q. V. Le. DropBlock: A regularization method for convolutional networks. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), MontrÃľal, Canada*, 2018.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV), Venice*, pp. 2980–2988, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 346–361. Springer International Publishing, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 9, pp. 1904–1916, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[12] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang. Mask Scoring R-CNN. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6409–6418, 2019.

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, France*, 2015.

[14] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu. A survey of deep learning-based object detection. *IEEE Access*, Vol. 7, pp. 128837–128868, 2019.

[15] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul*, pp. 8420–8429, October 2019.

[16] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA*, June 2019.

[17] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 25, January 2012.

[18] Q. Li, S. Jin, and J. Yan. Mimicking very efficient network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, pp. 7341–7349, July 2017.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, July 2017.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV), Venice*, 2017.

[21] S. Liu, D. Huang, and Y. Wang. Learning spatial fusion for single-shot object detection. *arXiv:1911.09516*, 2019.

[22] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT*, pp. 8759–8768, June 2018.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proceedings of the 14th European Conference on Computer Vision, Amsterdam, the Netherlands*, Vol. 9905, pp. 21–37, October 2016.

[24] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations (ICLR)*, April 2017.

[25] C. D. Manning, P. Raghavan, and H. Schütze. *An Introduction to Information Retrieval*, chapter 8.4. Cambridge University Press, 2009. http://www.informationretrieval.org/.

[26] D. Misra. Mish: A self regularized non-monotonic activation function. In *Proceedings of the 31st British Machine Vision Conference (BMVC)*, September 2020.

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel (ICML)*, 2010.

[28] R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *International Conference on Systems, Signals and Image Processing (IWSSIP)*, jul 2020.

[29] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[30] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, July 2017.

[31] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv:1804.02767*, 2018.

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 28, 2015.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, Vol. 323, pp. 533–536, October 1986.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy*, pp. 618–626, 2017.

[35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int'l J. Comput. Vis.*, Vol. 128, pp. 336–359, 2020.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826, 2016.
[39] M. Tan and Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, Long Beach, California*, 2019.
[40] M. Tan, R. Pang, and Q. V. Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA*, pp. 10778–10787, June 2020.
[41] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I-H. Yeh. CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA*, pp. 1571–1580, 2020.
[42] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu. Frustratingly simple few-shot object detection. In *Proceedings of the 37th International Conference on Machine Learning*, 13–18 Jul 2020.
[43] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.
[44] Z. Yao, Y. Cao, S. Zheng, G. Huang, and S. Lin. Cross-iteration batch normalization. *arXiv:2002.05712*, 2020.
[45] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul*, pp. 6022–6031, October 2019.
[46] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.
[47] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33, pp. 9259–9266, 2019.
[48] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-IoU Loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New York, USA*, Vol. 34, pp. 12993–13000, February 2020.

付録 A　Average Precision (AP) と mAP

物体検出の評価基準として一般的に用いられる average precision (AP) と mean average precision (mAP) を文献 [28] に沿って説明する．

A.1　IoU (Intersection over Union) と確信度 (confidence)

物体検出器の出力には，bbox 座標と各クラスの確信度が必ず含まれる．これらの値を用いて検出結果の正しさを評価する．

確信度 (confidence value) は多くの場合 0.0 から 1.0 までの確率を表す実数値で表現され，クラス確率，スコアなどと呼ばれることもあり，指定されたクラスに属する確率の推定値を表す．

IoU は bbox の一致の程度の指標であり，推定 (predicted) bbox が教師データ (ground truth) の bbox にどれくらい近いかを定量的に表したものである (図 7 参照)．

物体検出における正しい検出 (正検出) とは、推定 bbox と正解 bbox との IoU が閾値以上であり，且つ，推定 bbox に対してクラスが正しく予測された場合を言う．IoU が閾値より小さい場合，または，クラスを間違えている場合は，誤った検出 (誤検出) という．

A.2　適合率 (precision) と再現率 (recall)

表 7 は分類器の性能を数値化するときに使われる混同行列 (confusion matrix) である．物体検出の性能評価に活用するために，混同行列に対して物体検出の検出結果を次のように当てはめて考える：

TP: True Positive　正しく検出された個数
FP: False Positive　誤って検出された個数

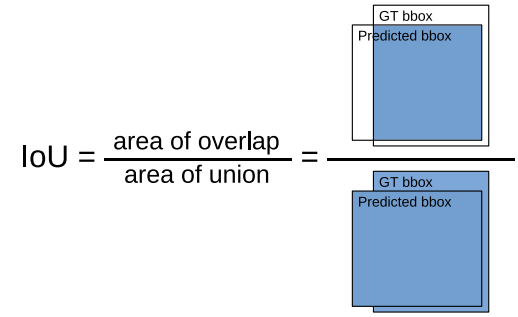

図 7　IoU (Intersection over Union). bbox の一致の程度を IoU (0≤IoU≤1) で測る．

表 7　混同行列 (confusion matrix).

		予測されたクラス P	予測されたクラス N
実際のクラス	P	真陽性 (TP)	偽陰性 (FN)
	N	偽陽性 (FP)	真陰性 (TN)

(a)　(b)　(c)

図 8　Bounding box の例 (猫の検出). (画像は https://unsplash.com/photos/Eoth0QkgT4I のものを使用．)

FN: False Negative　検出されなかった教師データ bbox の個数
TN: True Negative　定義しない (物体検出タスクでは，検出すべきでない bbox は無数に存在して数えられないため).

この定義より，全予測 bbox 数は TP+FP，全教師データ bbox 数は TP+FN で表される．

一つの教師データ bbox に対して複数の予測 bbox が検出された場合は，確信度が最大のものを TP とし，その他のものは FP とする．これは検出個数が多すぎることも誤りと捉えられるからである．

適合率 (precision)，再現率 (recall) は，TP，FP，FN を用いて次式で定義される：

$$\text{適合率 Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} = \frac{(\text{正しく検出された個数})}{(\text{全予測 bbox 数})} \quad (1)$$

$$\text{再現率 Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \frac{(\text{正しく検出された個数})}{(\text{全教師データ bbox 数})} \quad (2)$$

適合率，再現率はモデルの評価指標であり，両者はトレードオフの関係にある．構築したモデルの良し悪しをどちらを基準に評価するかは，モデルの利用目的に合わせて判断する必要がある．

A.3　PR 曲線 (Precision-Recall Curve)

PR 曲線とは，確信度の閾値を変化させて (再現率，適合率) をプロットしたグラフである (図 9)．以下では図 8 の検出結果が得られた場合を考える．5 匹の猫 (赤枠) に対して物体検出モデルが 10 個の bbox(緑枠) を検出したものとする．検出結果 (表 8) を確信度の大きい順に並べた表

表 8　物体検出結果.

図 8	Box	確信度	IoU	T	F
(a)	A	92%	0.78	1	0
(a)	B	88%	0.73	0	1
(a)	C	74%	0.32	0	1
(b)	D	83%	0.78	1	0
(b)	E	80%	0.86	0	1
(b)	F	89%	0.41	0	1
(b)	G	84%	0.91	1	0
(c)	H	96%	0.86	1	0
(c)	I	78%	0.21	0	1
(c)	J	72%	0.66	1	0

表 9　物体検出結果 (確信度でソートして precision と recall を計算したもの).

図	Box	確信度	IoU	T	F	TP	FP	TP+FP	TP+FN	Precision	Recall
(c)	H	96%	0.86	1	0	1	0	1	5	1.00	0.2
(a)	A	92%	0.78	1	0	2	0	2	5	1.00	0.4
(b)	F	89%	0.41	0	1	2	1	3	5	0.67	0.4
(a)	B	88%	0.73	0	1	2	2	4	5	0.50	0.4
(b)	G	84%	0.91	1	0	3	2	5	5	0.60	0.6
(b)	D	83%	0.78	1	0	4	2	6	5	0.67	0.8
(b)	E	80%	0.86	0	1	4	3	7	5	0.57	0.8
(c)	I	78%	0.21	0	1	4	4	8	5	0.50	0.8
(a)	C	74%	0.32	0	1	4	5	9	5	0.44	0.8
(c)	J	72%	0.66	1	0	5	5	10	5	0.50	1.0

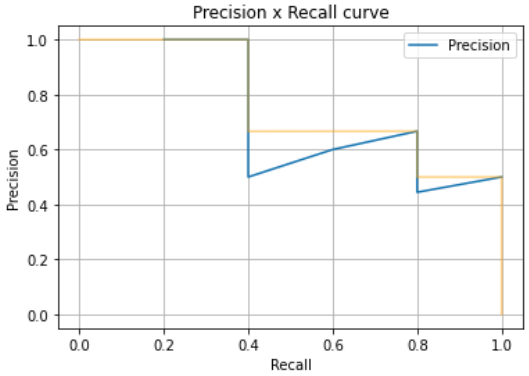

図 9　PR 曲線 (青) とそれに対応する $P_{\text{interp}}(R)$(黄).

(表 9) を作成して，(再現率，適合率) の推移 (右端の 2 つのカラム) を求める．表 9 の recall と precision をグラフにプロットしたのが図 9 の青線である．PR 曲線は，曲線が右上にあるほど性能が良いことを示す．

A.4　AP と mAP

PR 曲線をもとに precision の平均を求めたものが，ほぼ average precision (AP) になるが，正確には次式で定義する interpolated precision $P_{\text{interp}}(R)$ を用いて AP を計算する：

$$P_{\text{interp}}(R) = \max_{\tilde{R}:\tilde{R}\geq R} P(\tilde{R}).$$

この式は，その R より右側の $P(R)$ の最大値を値として持つという意味であり，ちょうど左上から水を流して水たまりの水面を表すと考えてよい (図 9 の黄色の線)．AP はこの $P_{\text{interp}}(R)$ の下の領域の面積として定義される．

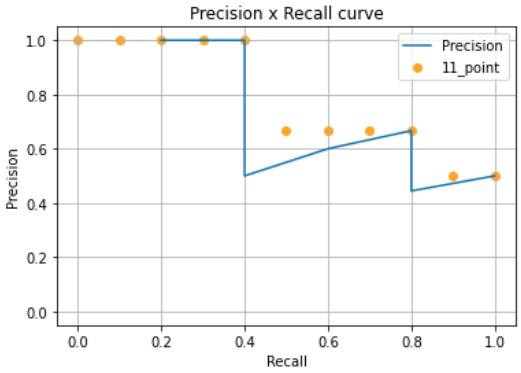

図 10　P_{interp} 曲線 (11 点の代表点による).

そもそも precision と recall は背反する関係にあるため，曲線全体では，閾値を調整してどちらかを増やそうとするともう一方が減少する．しかしながら，狭い範囲ではガタガタしており，もし R (recall) を増加させることで $P(R)$ (precision) も増加するときには増加したところを採用して AP を計算すべきという解釈である [25].

AP の計算方法には，次の 2 つの計算方法がある：

All-point interpolation AP は $P_{\text{interp}}(R)$ の下の領域の面積を厳密に計算したものである：

$$\text{AP}_{\text{all}} = \int_0^1 P_{\text{interp}}(R)\mathrm{d}R.$$

11-point interpolation AP は PR 曲線を $\{0, 0.1, 0.2, \cdots, 1.0\}$ の 11 個の recall レベルでの precision で代表させ，それらの平均値を計算したものである (図 10)：

$$\text{AP}_{11} = \frac{1}{11} \sum_{R\in\{0,0.1,0.2,\cdots,1.0\}} P_{\text{interp}}(R).$$

AP50, AP75, AP@[.50:.05:.95].　AP の値は IoU の閾値に依存する．閾値 X を指定した AP はしばしば APX と記される．AP50 は IoU=.50，AP75 は IoU=.75 を表す．AP@[.50:.05:.95] は，ステップサイズを 0.05 とし，IoU=.50 から IoU=.95 までのそれぞれの IoU 閾値で算出した AP の平均を計算したものである．単に AP と書いた場合，この AP@[.50:.05:.95] を表す場合があるので注意が必要である．

mAP (mean average precision).　通常，AP は特定のクラス (上記の猫など) に対する性能を表すが，検出対象の全クラス (猫，犬，車，$\cdots$) に対して AP の平均を計算したものを mAP といい，次式で定義される：

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i.$$

ただし，AP_i は i 番目のクラスに対する AP，N は評価すべきクラスの総数である．