原著論文 Original Paper

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 Junya Kaneko
Morning Project Samurai 株式会社
Morning Project Samurai Inc.
junya@mpsamurai.com, http://www.mpsamurai.com

山田 貢己 Miki Yamada
（同　　上）
m.yamada@mpsamurai.com

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

Summary

「ショートノート」は 200 ワード，それ以外は 200〜500 ワード以内の英文で summary を記す (ここは，論文執筆後に書く．)

1. ま え が き

この論文の狙い: このサーベイは，理工系大学 2 年生程度の数学の知識を前提に，物体検出をこれから始めるにはどうすればよいかという道すじを伝えることを目的として執筆したものである．物体検出でできることは何?から始まり，物体検出をするために必要なもの (ハード，ソフト，データ，知識，明確な目的) を簡潔に纏めてある．また，読者にとっての理想的なサーベイ (即時性，分かりやすさ，一言で説明，参考文献は充実) というものの一つの解として，随時更新される GitHub 上の (日本語で書かれた) 物体検出まとめサイトとライブラリを紹介する．

世の中の状況: 近年,「AI(人工知能)」という言葉が国内外に蔓延しており，技術者のみならず一般の人の日常生活にもすっかり浸透した．常に手の届くところに AI があり，AI に囲まれて生活していると言っても過言ではない．テレビやインターネットの画像は，本物と見間違えるほどの人工画像で溢れ，スマホや机上のスピーカー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ようになった．高速道路を自動運転する車も増えている．

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進めて，「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，犬，猫，自動車，飛行機，...) 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，これに数時間から数日要すると言われている．

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

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.

		物体検出	
		する	しない
画素毎の	全画素	Panoptic segmentation [Kirillov 19]	Semantic segmentation
クラス分類	物体領域のみ	Instance segmentation (YOLACT++ [Bolya 20], MS R-CNN [Huang 19] など)	—
	しない	物体検出 (YOLOv4 [Bochkovskiy 20], EfficientDet [Tan 20] など)	—

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)

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，並びに，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).

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.

明確な目的 何がしたいのかということを明確化しておくことがが，物体検出を行おうとするときに重要となる．物体検出は新しい技術であり，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くなりがちである．他の方法では解決できないのか？と問いかけて，本当にこれが必要であることを確認しておくべきである.

理想的なサーベイとは: 最新技術のサーベイ論文は，有用であり様々な分野で昔から (論文雑誌が生まれた頃から) 活用されていると思われる．しかしながら，進歩の速い分野においてはサーベイが出た頃には既に内容が古くなってしまっているという問題が往々にして起こる．また，とても良く書かれたサーベイほど内容が濃く多くなり，執筆に時間と労力を要するのはもちろん，それを読み解くのにも時間を要するということもよくある.

我々は，github 上に随時更新される形式でサーベイを公開することを試みた．出版されたときには既に古くなっているという懸念を取り払える可能性を期待している．また，この分野に新規参入しようとしている人がなるべく短時間で必要な情報にたどり着き，取り組んでいる問題を解決する最適な方法を見つけたり，或いは，新たな研究に取り組めることを目指し，内容の拡充性や緻密性よりも，なるべく視覚的に解りやすいコンパクトな内容になるよう心掛けた.

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) [Bochkovskiy 20], EfficientDet(検出精度が高い) [Tan 20] 等の検出器を使用する.

Instance segmentation を行わせる場合は，YOLACT++ (高速) [Bolya 20]，MS R-CNN(検出精度が高い) [Huang 19] 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 [Kirillov 19] を使用する．これは，物体検出器と 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性能向上の可能性がある.

2·2 使えるデータやリソースに制約がある場合 (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.

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
Few-Shot Object Detection with Attention-RPN and Multi-Relation Detector		Few-shot 物体検出	• ★作成予定.	• ★作成予定.
Two-stage Fine-tuning Approach (TFA)	[Wang 20b]	Few-shot 物体検出	• Few-shot 物体検出のための 2 ステージ fine-tuning を提案．最終層のみ fine-tuning する. • Faster R-CNN がそのまま使われている (few-shot 学習方法が新しい). • 特徴抽出器は backbone(ResNet, VGG16 等)，RPN，FC sub-network を含む.	• メタ学習の従来手法よりも 2〜20 ポイント優れている．ただし，少数サンプルの分散が大きいと信頼性が低下する.
Feature mimic method	[Li 17]	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	[Selvaraju 17, Selvaraju 20]	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.

学習データが少ない場合 (数個〜十数個) には，Few-shot 物体検出の方法が必要になる [Wang 20b].

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 [Li 17].

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 [Selvaraju 17, Selvaraju 20] 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，推論実行に必要なデータに正解データを加えたものであり，いわゆる教師あり学習を行わせる．ただ，推論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 [Jiao 19]：

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．Faster

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	[He 16]	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)	[Lin 17a]	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	[Ren 15]	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	[Lin 17b]	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)	[Redmon 18]	物体検出	• YOLOv2 の改良 (one-stage detector). • 320 × 320 YOLOv3 は 22 ms で走り，SSD 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)	[Tan 20]	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)	[Bochkovskiy 20]	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 65 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

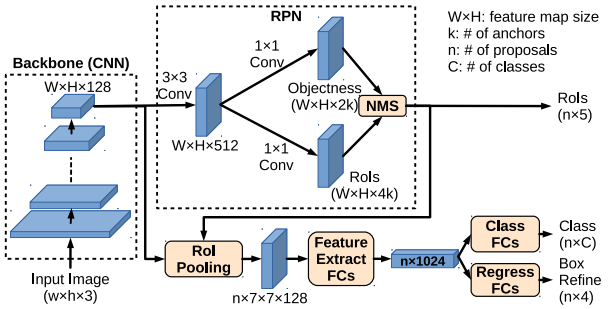


図 1 Faster R-CNN の構造.

R-CNN では Region Proposal Network (RPN) と呼ばれている.

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

One-stage 検出器 推論速度が高速である (YOLO, SSD 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向上している.

3·2 Two-stage 検出器

§1 Faster R-CNN

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 [Ren 15]. 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 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 [Simonyan 15] 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.

§2 Two-Stage Fine-Tuning Approach (TFA)

Few-shot 物体検出のためのシンプルな two-stage の fine-tuning 法である [Wang 20b]. 元来，深層学習は膨大な個数の学習データが必要であり，実用上の観点では学習データを集めることが困難な場合があった. それゆえ，少ないサンプルから珍しいオブジェクトを見つけることを目的とする few-shot 物体検出の開発が進められている。few-shot 学習の方法としてはメタ学習が有望と見られてきたが，この TFA はそれまで注目されてこなかった fine-tuning に焦点をあてることで大きな性能向上をもたらした. 最終層のみ fine-tuning することが few-shot 物体検出にとって重要であることを発見し，それを用いた Two-stage Fine-tuning Approach (TFA) を提案した. メタ学習の従来手法よりも 2〜20 ポイント高い精度を出している. ただし，少数サンプルの分散が大きいと信頼性が低下すると言われている. ベースとなる検出モデルは，Faster R-CNN がそのまま使われている. 本手法はネットワーク構造には殆ど依存せず，few-shot 学習方法に特徴がある.

Two-stage fine-tuning approach (TFA) の学習:

(1) Base model training (1st stage)

特徴抽出器 $\mathcal{F}$ と box 予測器 (分類器, 回帰器) を base クラスの学習セット C_b で学習する. そのときの損失関数は Faster R-CNN [Ren 15] と同じ

$$L = L_\mathrm{rpn} + L_\mathrm{cls} + L_\mathrm{loc}$$

を用いる. L_rpn は RPN の出力に適用され，前景を背景から区別し，anchor を学習データに近づける. L_cls は box 分類器 C に対するクロスエントロピー損失，L_loc は回帰器 R に対する平滑化 L1 損失である.

(2) Few-shot fine-tuning (2nd stage)

バランスのとれた 1 クラスあたり K shots の学習セッ

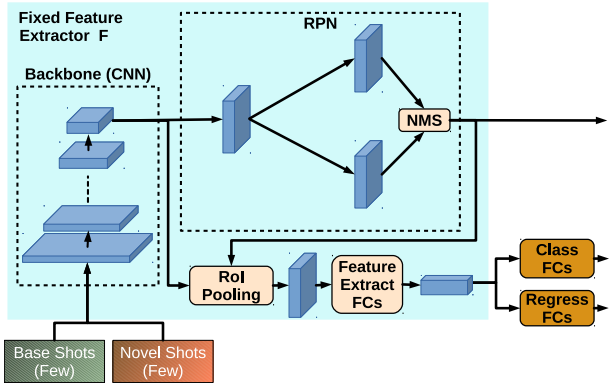


図 2 TFA の Few-shot fine-tuning (2nd stage).

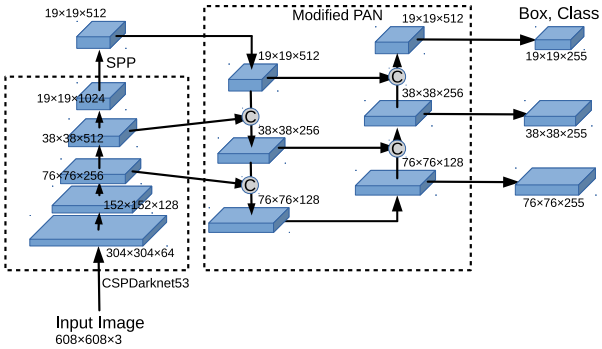


図 3 YOLOv4 の構造.

トを用意する．これらは base クラス C_{b} と novel クラス C_{n} を含む．Box 予測器の重みは乱数初期化する．特徴抽出器 $\mathcal{F}$ を固定して Box 分類器と回帰器 (検出モデルの最終層) を fine-tuning する．1st stage と同じ前記の損失関数 L を用い，学習率は 1st stage から 20 だけ減じる．また，2nd stage では cosine 類似度を用いて分類器の損失を正規化する工夫を取り入れている．

§3 Few-Shot Object Detection

3·3 One-stage 検出器

§1 YOLOv4

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 [Bochkovskiy 20]．YOLOv3 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(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))，neck(FPN や PAN のように、特徴量ピラミッドを昇り降り横断する処理)，head(「backbone + neck」が出力する特徴量から、クラス信頼度と bounding box オフセットを推論するネットワーク) 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 bounding box 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 bounding box 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，重みの更新のタイ

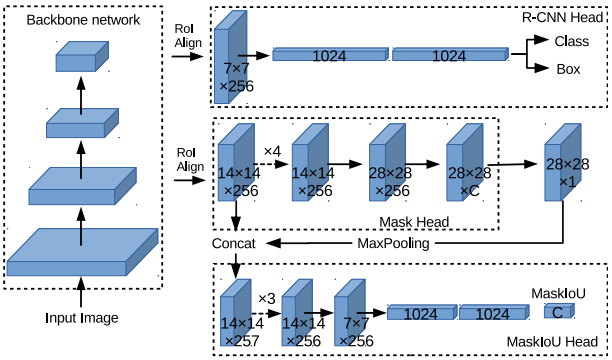


図 4 Mask Scoring R-CNN の構造.

ミングで，bias と scale を更新するようにしたものである．

Self-adversarial-training (SAT) まず，物体を写した元画像を改変し，物体が存在しない画像であると騙す画像を生成する．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 コサイン関数の半周期の形状を利用して，学習率を少しずつ減少させる．これを周期を増やしながら複数回繰り返す．

§ 2 **EfficientDet**

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)

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 [Huang 19]．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[He 17] に導入したモデルになっている (図 4)．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，推論時に予測 MaskIoU を分類スコアに掛け算して補正する．

§ 1 **MS R-CNN の学習**

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．

§ 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:

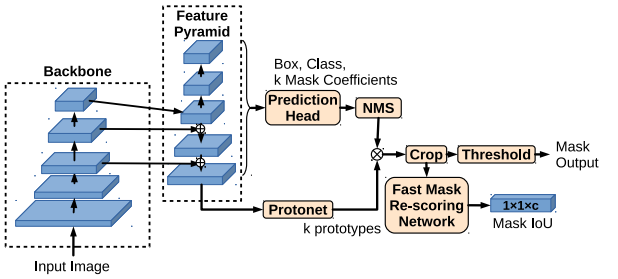


図 5 YOLACT++ の構造.

(1) R-CNN head が N 個の bounding box を出力する．
(2) N 個の bounding box のうち，SoftNMS[Bodla 17] で上位 k 個のボックスを選択する．
(3) 上位 k 個のボックスを 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．

4·2 YOLACT++

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 [Bolya 20]．fully-convolution モデルであり，deformable convolution を YOLACT[Bolya 19] 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[Lin 17b] 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 5)．

§ 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 Loss 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．

§ 2 **YOLACT++ の推論処理**

Prediction Head が各 anchor の Class 信頼度，bbox，k 個の mask 係数を出力し，Protonet が k 個の Prototype

*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.

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	[He 17]	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)	[Liu 18]	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)	[Huang 19]	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	[Bolya 20]	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	[Kirillov 19]	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

(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 を積和した結果 (全画面の mask) を，予測 bbox の外側を 0 で埋めたものを 2 値化して最終的な予測 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．

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

[Kirillov 19].

6. 物体検出に使われる性能向上技術

7. む　す　び

謝　辞

謝辞について

a

◇ 参 考 文 献 ◇

[Bochkovskiy 20] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection, *arXiv:2004.10934* (2020)

[Bodla 17] Bodla, N., Singh, B., Chellappa, R., and Davis, L. S.: Soft-NMS — Improving Object Detection with One Line of Code, in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5562–5570 (2017)

[Bolya 19] Bolya, D., Zhou, C., Xiao, F., and Lee, Y.: YOLACT: Real-time Instance Segmentation, in *International Conference on Computer Vision (ICCV 2019), Seoul*, pp. 9156–9165 (2019)

[Bolya 20] Bolya, D., Zhou, C., Xiao, F., and Lee, Y.: YOLACT++: Better Real-time Instance Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, No. DOI: 10.1109/TPAMI.2020.3014297 (2020)

[Ghiasi 18] Ghiasi, G., Lin, T.-Y., and Le, Q. V.: DropBlock: A Regularization Method for Convolutional Networks, in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montral, Canada* (2018)

[He 16] He, K., Zhang, X., Ren, S., and Sun, J.: Deep Residual Learning for Image Recognition, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016)

[He 17] He, K., Gkioxari, G., Dollár, P., and Girshick, R.: Mask R-CNN, in *2017 IEEE International Conference on Computer Vision (ICCV), Venice*, pp. 2980–2988 (2017)

[Huang 19] Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X.: Mask Scoring R-CNN, in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6409–6418 (2019)

[Jiao 19] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R.: A Survey of Deep Learning-Based Object Detection, *IEEE Access*, Vol. 7, pp. 128837–128868 (2019)

[Kirillov 19] Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P.: Panoptic Segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA* (2019)

[Li 17] Li, Q., Jin, S., and Yan, J.: 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 (2017)

[Lin 17a] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S.: Feature Pyramid Networks for Object Detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125 (2017)

[Lin 17b] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P.: Focal Loss for Dense Object Detection, in *2017 IEEE International Conference on Computer Vision (ICCV), Venice* (2017)

[Liu 18] Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J.: 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 (2018)

[Loshchilov 17] Loshchilov, I. and Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts, in *5th International Conference on Learning Representations (ICLR)* (2017)

[Redmon 18] Redmon, J. and Farhadi, A.: YOLOv3: An Incremental Improvement, *arXiv:1804.02767* (2018)

[Ren 15] Ren, S., He, K., Girshick, R., and Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in *Advances in Neural Information Processing Systems (NIPS)*, Vol. 28 (2015)

[Selvaraju 17] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D.: 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)

[Selvaraju 20] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, *Int'l J. Comput. Vis.*, Vol. 128, pp. 336–359 (2020)

[Simonyan 15] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, in *International Conference on Learning Representations (ICLR)* (2015)

[Tan 20] Tan, M., Pang, R., and Le, Q. V.: 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 (2020)

[Wang 20a] Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., and Yeh, I.-H.: 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)

[Wang 20b] Wang, X., Huang, T., Gonzalez, J., Darrell, T., and Yu, F.: Frustratingly Simple Few-Shot Object Detection, in *Proceedings of the 37th International Conference on Machine Learning* (2020)

[Woo 18] Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S.: CBAM: Convolutional Block Attention Module, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19 (2018)

[Yun 19] Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., and Choe, J.: CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South)*, pp. 6022–6031 (2019)

[Zheng 20] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D.: 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 (2020)

〔担当委員：××○○〕

19YY 年 MM 月 DD 日　受理

◇ 付　　録 ◇

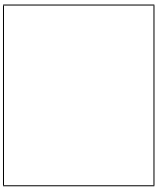
A. 付録のタイトル 1

付録の本文 1

表 5 物体検出の性能向上技術.

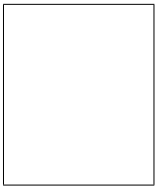
モデル名称	文献	用途	概要	特徴
CSPNet	[Wang 20a]	CNN の軽量化	• Cross Stage Partial Network (CSPNet) を提案し，従来法を軽くする. • ImageNet データセットにおいて，同等以上の精度で計算量を 20%削減．MS COCO の物体検出データセットでは AP50 に関しては最高性能を達成. • CSPNet は簡単に実装できて，ResNet, ResNeXt, DenseNet に基づくアーキテクチャを扱える汎用性を持つ.	
Convolutional Block Attention Module (CBAM)	[Woo 18]	CNN の性能向上.	• 任意の CNN に適用可能な attention module. • 分類と検出の評価実験において一貫して性能向上を示しており，オーバーヘッドは殆ど無い. • 可視化するときは、Grad-CAM を使用.	• 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.
DropBlock	[Ghiasi 18]	学習性能の向上	• 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. • 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.	
CutMix	[Yun 19]	データ拡張	• 学習画像間で、画像のパッチをカット＆ペーストし，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.
SGDR (Cosine annealing scheduler)	[Loshchilov 17]	学習性能の向上	• 周期的に SGD の warm restart をすることを提案．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. • YOLOv4 で使われている Cosine annealing scheduler はこれのこと.	• 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.
Distance-IoU (DIoU) 損失／ Complete IoU (CIoU) 損失	[Zheng 20]	学習性能の向上	• 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) 損失を提案する. • DIoU と CIoU を，YOLOv3, SSD, 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.	

著 者 紹 介



金子 純也（正会員）

著者 1 の略歴



山田 貢己（正会員）

1989 年東京大学大学院物理学専攻修了．理学博士．同年株式会社東芝入社．ニューラルネットワークの研究開発，セキュリティ技術，画像認識技術，テレビの高画質化技術，車載画像認識プロセッサ等の開発業務に従事．2020 年ジャパニアス株式会社に入社．現在，Morning Project Samurai 株式会社において AI 開発業務に従事．