Objectnessを導入したSSDによる未知クラスアイテムの認識

○荒木諒介 †，長谷川昂宏 †，山内悠嗣 †，山下隆義 †，藤吉弘亘 †，橋本学 ‡，堂前幸康 ††

○ Ryosuke ARAKI †，Takahiro HASEGAWA †，Yuji YAMAUCHI †，
Takayoshi YAMASHITA †，Hironobu FUJIYOSHI †，Manabu HASHIMOTO ‡ and
Yukiyasu DOMAE ††

†：中部大学，{ ryorsk@mprg, tkhr@mprg, yuu@isc, yamashita@cs, hf@cs }.chubu.ac.jp
‡：中京大学，mana@isl.sist.chukyo-u.ac.jp
††：三菱電機株式会社，Domae.Yukiyasu@cb.MitsubishiElectric.co.jp

＜要約＞本稿では，Objectnessを導入したSingle Shot Multibox Detector (SSD) による未知クラス物体に対応した物体検出法を提案する．提案手法は，物体検出アルゴリズムであるSSDに「物体らしさ」を表すObjectnessを導入する．これにより，学習データに含まれていないアイテム（未知クラスアイテム）に対して，物体であるか，そうでないかの認識が可能となる．Amazon Robotics Challenge (ARC) のために作成したデータセットを用いた評価実験により，提案手法は49.59%の未知クラスアイテムを検出することを確認した．
＜キーワード＞物流ロボット, Amazon Robotics Challenge, Robot Vision, Deep Learning, Dataset

1　はじめに

ロボット技術の進化により，eコマースにおける物流倉庫の自動化が進んでいる．その一例として，顧客から商品の注文がデータベースに入力されたとき，ロボットが商品の入っている棚を持ち上げてピッキングオペレータの元まで自動搬送するシステムがある [1]．現在，この商品のピッキングは人手によって行われているため，ピッキングロボットによる自動化が期待されている．ロボットが商品をピッキングするためには，画像から物体検出，把持位置検出を行う．特に，物体検出タスクにて正しい物体を検出することは，注文を受けた商品を正しく顧客へ届けるための最初の重要な処理である．しかし，物流倉庫では日々新しい商品が追加されるため，認識システムへのデータ登録に手間がかかる．さらに，認識システムに機械学習を用いる場合，学習用データの作成が必要となる．このような背景の下，物流の自動化技術を競うロボット大会「Amazon Robotics Challenge (ARC) 」では，競技開始直前に追加される新しいアイテムをピッキング対象とする課題も含まれている．本研究では，学習データに含まれていない新しいアイテムの検出に対応するために，物体らしさを同時推定する手法を提案する．本手法はSSDをベースとし，学習データに含まれていないアイテム，すなわち未知クラスアイテムに対して物体であるか，そうでないかの検出を実現する．

2　関連研究

これまでに提案されている物体検出手法の中でも，高精度かつ高速な手法としてFaster R-CNN[2], YOLO[3]，SSD[4] がある．また，物体検出手法よりも細かく，ピクセル単位でのクラス分類を行うことができるセマンティックセグメンテーションアルゴリズムの1つとしてSegNet[7] がある．本章では，これらの関連研究について述べる．

2.1　Faster R-CNN

Faster R-CNNでは，画像内に物体と予測される矩形を検出し，その矩形の中にある物体が何かを認識するクラス分類を個別で行う．従来のR-CNN[5] およびFast R-CNN[6] と比較すると，検出アルゴリズム全体がCNNで作られているため高精度かつ高速な検出ができる．

まず，Region Proposal Network (RPN) を用いて物

体と予測される矩形を検出する．RPN は，特徴マップ上の Sliding Window により物体の有無を計算する．Windows ごとに Anchor と呼ばれるアスペクト比が固定された k 個の矩形があり，Anchor ごとに物体らしさのスコアを計算する．これにより，様々な形の物体候補領域を検出することができる．

次に，RPN で出力された物体候補領域を RoI Pooling を通して，全結合層に入力する．これにより，物体候補領域の物体が何の物体かを識別することができる．

2.2 You Only Look Once(YOLO)

YOLO は Faster R-CNN と違い，画像を入力するとその画像から複数の矩形の検出とクラス分類を同時に行う．各矩形に対して物体らしさの確率を出力しているため，出力した物体らしさ確率が高い検出結果を出力する．YOLO による物体検出は非常に高速だが，検出精度は Faster R-CNN より低い．

YOLO は画像を 448×448pixels にリサイズし，$S \times S$ のグリッドに分割する．このグリッドの 1 つをグリッドセルと呼ぶ．各グリッドセルは矩形の情報と，その矩形の信頼スコア (confidence scores) を持っている．矩形の情報は，中心座標 (x, y)，幅 w，高さ h の 4 つである．矩形の信頼スコアは，推定結果と教師信号の重なり率を示す Intersection over Union (IoU) と物体らしさのスコアを掛けた値となっている．グリッドセルにオブジェクトがなかった場合は，信頼スコアは 0 となる．また，各グリッドセルはすべてのクラスに対する尤度 $C = \Pr(\text{Class}_i|\text{Object})$ も同時に出力する．ただし，矩形の数に限らず，C は 1 組だけである．

最後に，confidence scores が高かった矩形を物体領域として検出し，その矩形が属するグリッドセルのクラス尤度が高いものを矩形が推定するアイテムのクラスとする．

2.3 Single Shot Multibox Detector(SSD)

SSD は YOLO と同じく，物体候補領域の検出とクラス分類を単一のネットワークで行うことができる．精度は Faster R-CNN と同程度であるが，速度は Faster R-CNN よりも SSD の方が明らかに高速である．物体候補領域の検出には Default box を用いる．これは Faster R-CNN で使用されている Anchor と似ているが，Default Box は複数の解像度の特徴マップで適用する．

ただし，この手法では Default box の数が大きく増える．そのため，box のほとんどがアイテムの候補領域でない Negative box であり，アイテムの候補領域である Positive box に比べて遥かに多いため，不均衡な状態となる．この対策として，Hard negative mining を行う．Negative box のうち教師信号の box との重なり率が低いものを学習させないようにすることで，Positive box と Negative box の比を 1:3 にする手法である．これにより，安定した学習を行うことができる．

2.4 SegNet

SegNet は，Encoder とそれに対応する Decoder からなるセグメンテーション手法である．Encoder および Decoder は畳み込み処理，バッチ正規化，活性化関数からなる．Encoder は活性化関数の出力に対してプーリングを行い，プーリングで選択された画素の座標を記憶する．Decoder はプーリングで記憶した座標を用いて，畳み込みの前にアップサンプリングを行う．Encoder で畳み込んだ特徴マップを Decoder により元の画像サイズまで戻したのち，Softmax 分類器によりピクセル単位の分類を行う．これにより，高解像度なセグメンテーションを行うことができる．

物体検出タスクにセマンティックセグメンテーションを適用することにより，物体同士の境界が推定できるようになり，物体の重なりを考慮した把持位置検出や行動計画を行うことができる．

3 ARC2017 RGB-D Dataset

本研究では，ARC2017 向けに作成したデータセットを用いる．このデータセットは ARC2017 の競技で使われた 40 種類を赤色の箱 (Tote) に入れて撮影した画像からなる．アイテムは様々な形状や属性を持ち，箱状のアイテムやビニールで梱包されたアイテム，非剛体のアイテムなどがある．1 シーンあたり 1 から 8 個程度のアイテムが含まれており，アイテム同士が重ならないように配置されたシーン，重なりを考慮せず配置したシーン，アイテムを画像内に 1 つだけ配置したシーンがある．すべてのシーンに RGB 画像，距離画像，アイテムが矩形で囲まれたバウンディングボックス，アイテムの領域を示すセグメンテーションラベルが含まれている．このデータセットは Faster R-CNN などの物体検出タスクや，SegNet[7] などのセマンティックセ

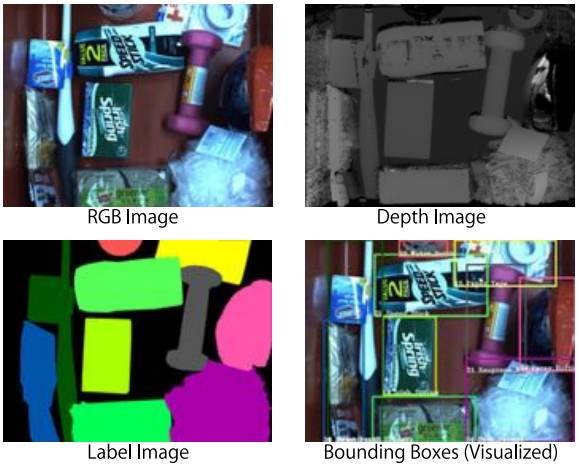

図 1　データセットに含まれる画像と教師ラベル.

グメンテーションタスクの学習ができる．画像のサンプルを図 1 に示す．また，本データセットは我々の研究グループである Machine Perception and Robotics Group の Web ページ [1] にて公開している．

3.1　RGB-D 画像

データセットには 1280 × 960pixels の RGB 画像と，そのシーンに対応する距離画像が含まれている．40 種類のアイテムを Tote に複数個入れて撮影したシーンが 1000 枚 (500 シーンの Tote を 2 回撮影)，Tote にアイテムを 1 つだけ入れて撮影したシーンが 410 枚ある．また，複数個アイテムのある画像のうち 800 枚を学習用，200 枚を評価用に分けている．

3.2　バウンディングボックス

すべての RGB 画像にはバウンディングボックスのアノテーションがつけられている．アノテーションファイルは画像ごとに分かれており，ボックスの座標とアイテム ID が記述されているテキストデータである．アイテム ID はアイテムごとにつけられた番号 (1〜40) である．ただし，ID:0 は背景クラスとするためバウンディングボックスは存在しない．アノテーションは人が画像を見た時に確認できる範囲のみつけられているため，隠れが発生している場合は見える範囲のみボックスで囲まれている．強い隠れにより，どのアイテムか全く判別することができない場合はボックスをつけていない．

3.3　セグメンテーション画像

セマンティックセグメンテーションによって物体同士の境界を推定すると，重なりを考慮した把持位置検出や行動計画を行うことができる．このデータセットには，セマンティックセグメンテーションの学習および評価を行うために，RGB 画像をアイテム領域ごとにピクセル単位で色分けしたセグメンテーション画像が含まれている．アイテムごとに異なる色で塗り分けられ，背景は黒色で塗られている．

3.4　評価ツール

本データセット専用の評価ツールも合わせて公開している．本ツールは，教師信号と検出結果の IoU を用いることにより検出に成功したかを判定する．ツールに検出結果と正解ラベルを入力すると，認識率 (未検出を除く認識率)，未検出率 (未検出の割合)，平均 IoU (誤検出を含めたすべての IoU の値の平均) および Confusion Matrix を計算し出力する．本ツールでは，IoU を式 (1) で計算する．

$$\frac{R_d \cap R_t}{R_d \cup R_t} \tag{1}$$

ここで，R_d は検出したボックスの領域，R_t は教師信号の矩形領域である．

3.5　ベンチマークテスト

本データセットを用いて，Faster R-CNN，YOLO および SSD の評価実験を行う．Faster R-CNN は Chainer CV[2] に実装されているコード，SSD は自作したコードを用いた．SSD は色調変化，反転，切り出しによる 30 倍の Data augmentation，2 エポックに 1 回の Hard negative mining を行った．また，SSD は高速化のために C++で書き直したコードを用いてテストを行った．評価には GPU (GTX 1070) を用いて，各手法は評価ツールを用いて評価する．各手法の実験条件および評価結果を表 1 に示す．最も認識率が良かった手法は Faster R-CNN であり，次に SSD である．しかし，Faster R-CNN の検出速度は 2FPS と遅い．一方，SSD は 45FPS と高速である．

続いて，SegNet の評価実験を行う．学習およびテストともに ChainerCV に実装されているコードを用いた．評価結果を表 2 に示す．また，セマンティックセグメンテーション結果の例を図 2 に示す．背景クラスは概ね良好に認識できているが，アイテムの上に他のアイテムが置かれているケースで認識が失敗しやすい．

[1] http://mprg.jp/research/arc_dataset_2017

[2] https://github.com/chainer/chainercv

表 1　ベンチマークテストの実験条件と結果.

アルゴリズム	プログラム	iteration 回数	batchsize	認識率 [%]	未検出率 [%]	平均 IoU	速度 [FPS]
Faster R-CNN[2]	ChainerCV	700000	1	91.42	18.68	0.82	2
YOLO[3]	オリジナル	40000	64	56.72	40.57	0.75	15
SSD[4]	Chainer	223740	16	89.04	25.61	0.79	45

図 2　SegNet によるセマンティックセグメンテーション結果の例.

表 2　SegNet の結果.

iteration 回数	バッチサイズ	Global average accuracy	Class average accuracy	平均 IoU
60000	8	0.7819	0.7216	0.5397

アイテムの配置方法は無数にあるため，アイテムどうしの位置関係やアイテムそのものの配置されやすい場所などが学習により獲得しにくいため，このような結果になったと考えられる.

4　Objectness を導入した SSD

アイテムの正確な把持のためにはセマンティックセグメンテーションタスクで物体領域の境界を推定することが望ましい．しかし，セマンティックセグメンテーションで未知の物体を領域分割することは困難である．そこで，物体検出アルゴリズムのうちベンチマークテストで高精度かつ高速に検出できた SSD に対して，物体らしさを示す「Objectness」を導入する.

4.1　Objectness の識別器

Objectness を導入した SSD の構成は，物体候補領域の推定器とアイテムの識別器に加えて，Objectness の識別器を追加したものである．Objectness の識別器はアイテムの識別器と全く同じ識別器であり，「物体であるか，そうでないか」の 2 クラス分類を行う．このうち「物体である」尤度を Objectness と呼び，この値が閾値以上であれば物体があると判断する．物体であると判断されたボックスについてアイテム尤度を確認し，尤度が高いアイテム ID を最終的な検出結果とする．このとき，背景クラス (アイテム ID: 0) の尤度が高い場合は未知クラスアイテムとして判定する.

4.2　損失関数

Objectness を導入した SSD の損失関数は，オリジナルの損失関数に Objectness loss である $L_{obj}(x, o)$ を追

表 3　既知クラス物体と未知クラス物体の評価結果.

アルゴリズム	既知クラス 認識率 [%]	既知クラス 未検出率 [%]	既知クラス 平均 IoU	未知クラス 認識率 [%]	未知クラス 未検出率 [%]	未知クラス 平均 IoU
オリジナルの SSD	89.04	25.61	0.79	0	32.33	0.74
Objectness を導入した SSD	80.51	23.32	0.77	49.59	17.82	0.72

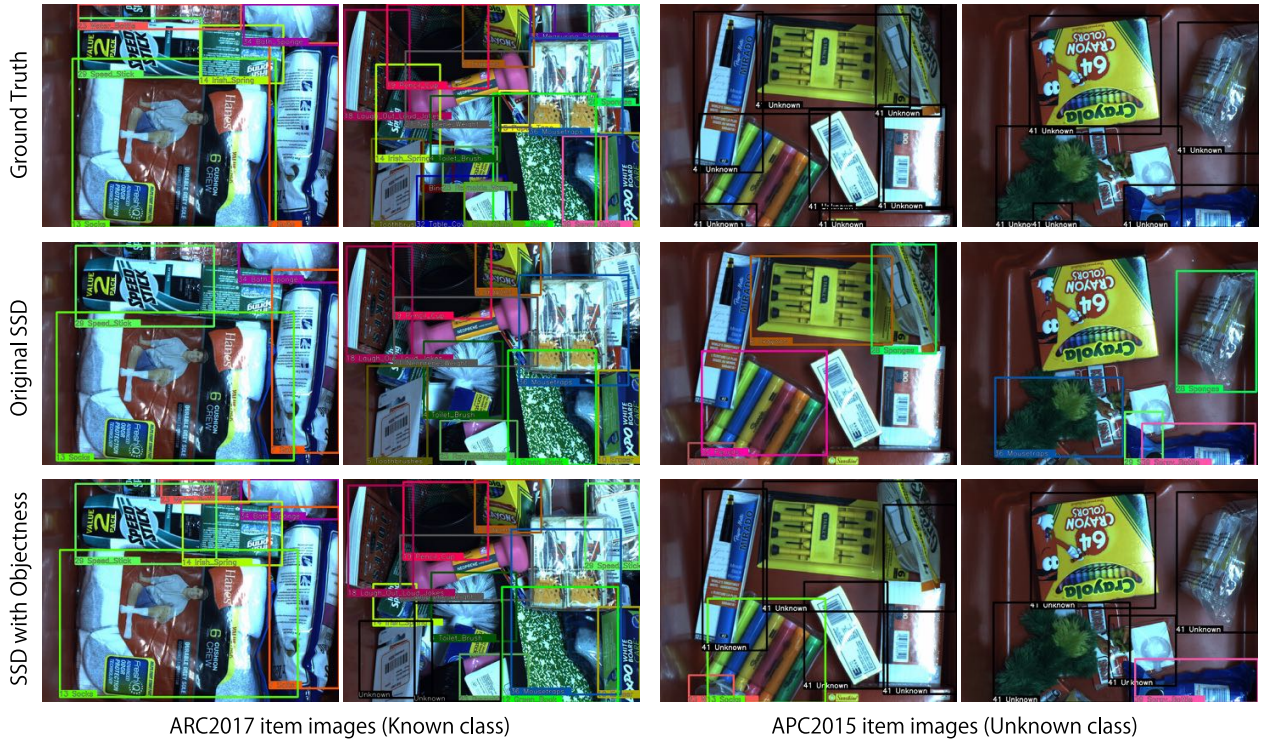

図 4　SSD と提案手法による検出結果例.

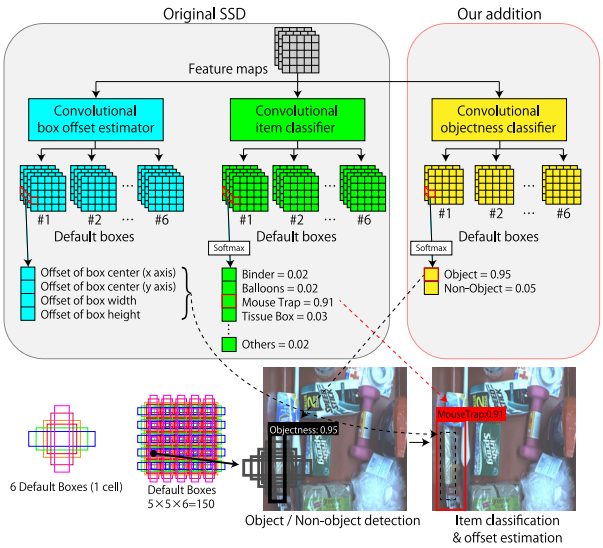

図 3　Objectness を導入した SSD の後段処理.

加している．この関数は式 (2) で表される．

$$L_{obj}(x,o) = -\sum_{i\in Pos}^{N} x_{ij}^{p}\log\left(\hat{o}_i^{p}\right) \sum_{i\in Neg}\log\left(\hat{o}_i^{0}\right)$$
$$\text{where } \hat{o}_i^{p} = \frac{\exp\left(o_i^{p}\right)}{\sum_p \exp\left(o_i^{p}\right)} \quad (2)$$

損失関数は式 (3) となる．

$$L(x,c,o,l,g) = \frac{1}{N}\left(L_{conf}(x,c)+L_{obj}(x,o)+\alpha L_{loc}(x,l,g)\right) \quad (3)$$

ここで，N はマッチした Default box の数である．$N = 0$ ならば損失は 0 となる．また，c は multiple classes confidences，o は objectness confidences，l は predicted box，g は grand truth box である．他はオリジナルの SSD と同じであるため，オリジナルの論文 [4] を参照されたい．

4.3　評価実験

Objectness の有用性を確かめるために，学習データにないアイテムを含むテスト用画像を用意した．テスト用画像に含まれるアイテムは APC2015 で使われた競技用アイテムである．ただし，APC2016 または ARC2017 で APC2015 のアイテムが使われていることがある．これについては除外する．このテスト用画像に対して認

識実験を行った．

オリジナルの SSD と Objectness を導入した SSD による，既知クラス物体と未知クラス物体の評価実験結果を表 3 に示す．表 3 より，既知クラスのアイテムは認識率が低下したが，未検出率は向上している．SSD では検出できなかったアイテムが Objectness の導入により検出できるようになったことがわかる．また，SSD では未知クラスのアイテムは全く認識できないが，提案手法では 49.59 % 認識可能であることがわかる．

図 4 に，SSD と提案手法による物体検出例を示す．オリジナルの SSD では未知クラスアイテムを認識できず，ボックスを検出できたとしても学習済みの似ているアイテムとして誤認識する．対して Objectness を導入した SSD は，既知クラスアイテムに似ているアイテムでなければ，一部の未知クラスアイテムについて認識できていることがわかる．

5 おわりに

本稿では，未知クラス物体の検出に対応するために Objectness を導入した SSD を提案した．我々の作成したデータセットを用いた評価実験により，未知クラス物体の検出が可能であることを確認した．今後は提案手法のさらなる精度向上と，ロボットシステムを考慮した把持位置の同時検出について研究する．

参考文献

[1] Amazon Robotics, “Vision”, [Online] `https://www.amazonrobotics.com/#/vision` (2018/1/16 参照).

[2] S. Ren, *et al.*, “Faster R-CNN: Towards real-time object detection with region proposal networks”, NIPS, pp.91-99, 2015.

[3] J. Redmon, *et al.*, “You Only Look Once: Unified, real-time object detection”, CVPR, pp.779-788, 2016.

[4] W. Liu, *et al.*, “SSD: Single Shot Multibox Detector”, ECCV, pp.21-37, Springer, 2016.

[5] R. Girshick, *et al.*, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR, pp.580-587, 2014.

[6] R. Girshick, “Fast R-CNN”, ICCV, pp.1440-1448, 2015.

[7] V Badrinarayanan, *et al.*, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”, arXiv preprint arXiv:1511.00561, 2015.