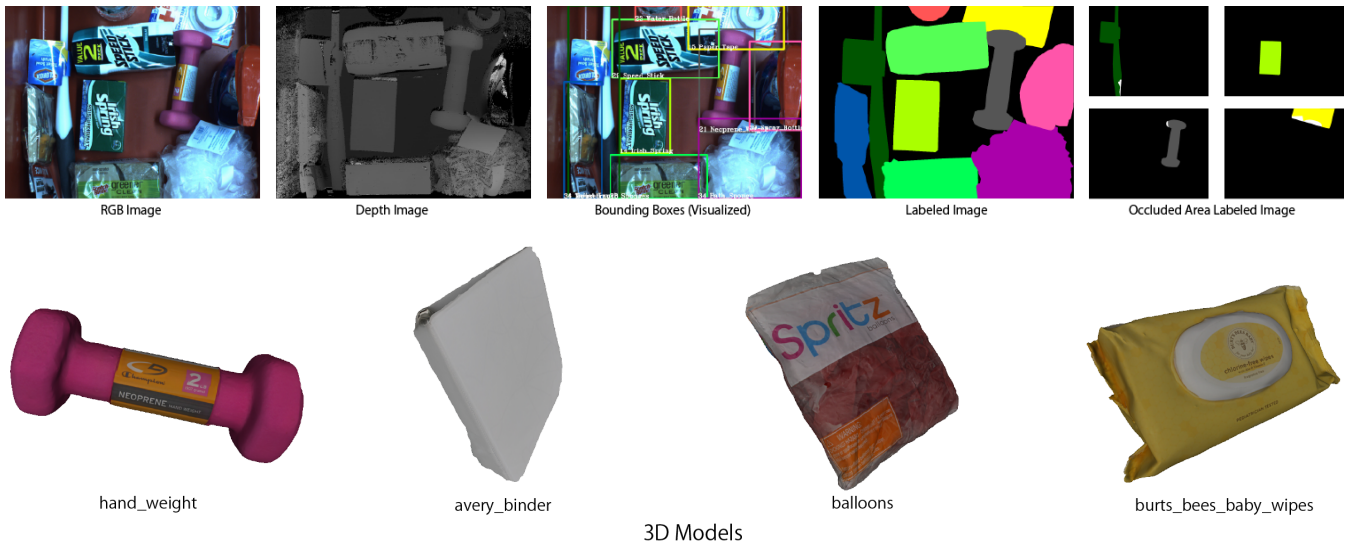


14 Sep 2017 Dataset Robotics Object Detection Semantic Segmentation

Team MC^2 : ARC2017 RGB-D Dataset

Amazon Robotics Challenge(ARC) では、棚(Storage)や箱(Tote)に陳列されたアイテムの中から指定されたアイテムをロボットが認識し、ピッキングする問題を扱っており、物流の自動化を目的としたコンペティションです。本Webページでは、Team MC^2がAmazon Robotics Challenge 2017用に作成したデータセットを公開します。データセットは三菱電機社製のMELFA-3D Visionで撮影し、RGB画像および距離画像の2種類のデータとなります。定量的評価のために、アイテム毎に指定された色でラベリングした正解画像、アイテムが重なっている部分を塗り分けたオクルージョン領域画像と各アイテムのバウンディングボックスの座標を記述したテキストファイルも用意しています。また、各アイテムをOrtery Technologies, Inc.製の3D MFPを用いて生成した3Dモデルも用意しています。



データセットの概要

アイテムは、ARCの競技用アイテム40種類です。箱状の形状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、様々な形状・属性のアイテムが用意されています。

下記の5つのタスクを想定してデータが用意されています。

Task1: 既知クラスアイテムのバウンディングボックス検出

Task2: 既知クラスアイテムのセグメンテーション

Task3: 既知クラスアイテムのオクルージョン領域のセグメンテーション

Task4: 未知クラスアイテムのバウンディングボックス検出

Task5: 未知クラスアイテムのセグメンテーション

データセットの構成

本データセットは、Toteの中にアイテムが複数ある状態、アイテムが1つだけある状態を想定して作成しています。機械学習・深層学習の学習と推論を想定して、学習用データ(train)と推論評価用データ(test_known)、未知アイテムの推論評価用データ(test_unknown)に分けています。

- ・ trainデータ: Toteにアイテムを複数入れ、Toteを2回にわけて撮影した画像(全440シーン×2回撮影=880枚)とToteのアイテムを1つだけ入れた画像(各アイテム10回撮影=410枚)
- ・ test_knownデータ: Toteにアイテムを複数入れ、Toteを2回にわけて撮影した画像(全110シーン×2回撮影=220枚)

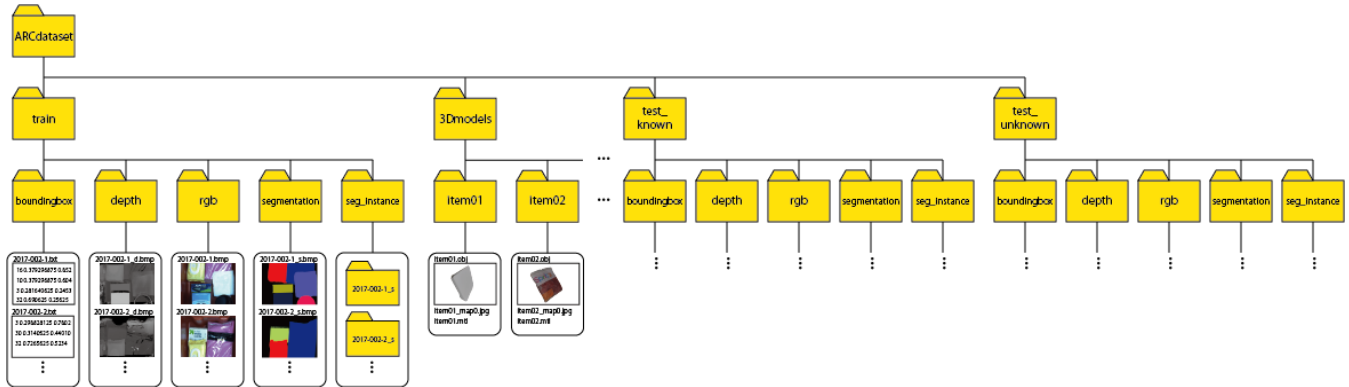
- ・test_unknownデータ:ToteにAPC2015のアイテムを複数入れ撮影した画像(全100シーン×1回撮影=100枚) APC2016・ARC2017で使われたアイテムは除外

2種類のシーンは、下記のようなフォルダ構成になっています。

trainデータと既知のtestデータと未知のtestでフォルダが分かれており、

さらにその中でバウンディングボックスデータ、RGB画像、距離画像、ラベリングデータ、オクルージョン領域ラベリングデータごとに分かれています。

3Dモデルデータはアイテムごとにフォルダが分かれています。



各データの形式

各データの形式は以下になっています。

RGB画像

- ・ファイル名 “[train|test_known|test_unknown]/rgb/2017-XXX-XX.bmp”
- ・画像サイズ “1280×960”

距離画像

- ・ファイル名 “[train|test_known|test_unknown]/depth/2017-XXX-XX_d.bmp”
- ・画像サイズ “1280×960”
- ・0～255のグレースケール画像で距離を表現

バウンディングボックスデータ

- ・ファイル名 “[train|test_known|test_unknown]/boundingbox/2017-XXX-XX.txt”
- ・アイテムIDとボックスの座標が記入されています。
- ・座標は「Box中心のX座標、Y座標、Boxの幅、高さ」の順に記述されています。また、0～1で正規化されています。
- ・未知アイテム(test_unknown)のアイテムIDは未知クラスを表す41に統一しています。

boundingbox/2017-xxx-x.txt








14	0. 386328125	0. 573958333333	0. 19609375	0. 385416666667
21	0. 7078125	0. 3890625	0. 2171875	0. 532291666667
29	0. 46640625	0. 222916666667	0. 4171875	0. 304166666667
4	0. 18515625	0. 500520833333	0. 2015625	0. 998958333333
Item ID	Center of Box (X)	Center of Box (Y)	Width of Box	Height of Box

ラベリングデータ

- ・ファイル名 “[train|test_known|test_unknown]/segmentation/2017-XXX-XX_s.bmp”
- ・画像サイズ “1280×960”

ラベリングデータはRGB画像1枚毎に作成しており、アイテムの領域に対して指定された色で塗り分けられています。また、背景は黒で塗りつぶされています。

ラベリングデータの各アイテムに対する番号はアイテムリスト"ItemList.pdf"に示しており、各アイテムのカラーコードはセグメンテーションカラーコードリスト"SegmentationTeach.xlsx"および下記テーブルに記しています。






Item ID	R	G	B	Color	Item Image
BG	0	0	0		
Item 1	85	0	0		
Item 2	170	0	0		
Item 3	255	0	0		
Item 4	0	85	0		
Item 5	85	85	0		
Item 6	170	85	0		
Item 7	255	85	0		
Item 8	0	170	0		
Item 9	85	170	0		

Item ID	R	G	B	Color	Item Image
Item 20	0	85	85		
Item 21	85	85	85		
Item 22	170	85	85		
Item 23	255	85	85		
Item 24	0	170	85		
Item 25	85	170	85		
Item 26	170	170	85		

Item ID	R	G	B	Color	Item Image
Item 10	170	170	0		
Item 11	255	170	0		
Item 12	0	255	0		
Item 13	85	255	0		
Item 14	170	255	0		
Item 15	255	255	0		
Item 16	0	0	85		
Item 17	85	0	85		
Item 18	170	0	85		
Item 19	255	0	85		

Item ID	R	G	B	Color	Item Image
Item 30	170	255	85		
Item 31	255	255	85		
Item 32	0	0	170		
Item 33	85	0	170		
Item 34	170	0	170		
Item 35	255	0	170		
Item 36	0	85	170		

					
Item 27	255	170	85		
Item 28	0	255	85		
Item 29	85	255	85		

					
Item 37	85	85	170		
Item 38	170	85	170		
Item 39	255	85	170		
Item 40	0	170	170		

オクルージョン領域ラベリングデータ

- ・ファイル名 “[train|test_known]/seg_instance/2017-XXX-XX_s/2017-XXX-XX_s_XX.bmp”
- ・画像サイズ “1280×960”
- ・各アイテムラベルにつき1枚データがあります。オクルージョン領域を白(255, 255, 255)で塗り分けています。

3Dモデル

- ・ファイル名 “3Dmodels/itemXX/itemXX.obj”

objファイルはWavefront Technologiesが開発したソフトウェア「The Advanced Visualizer」用のファイルフォーマットです。ポリゴンデータがアスキーで記述されており、現在では多くの3Dソフトがサポートしています。

Windows 10では3D Builder、macOSではXcodeで閲覧することができます。また、フリーソフトのMeshLabやBlender等でも閲覧することができます。

- ・マテリアルデータ “itemXX.mtl” およびテクスチャデータ “itemXX_mapX.jpg” が付随
- マテリアルデータとテクスチャデータはobjファイルと同じディレクトリに置いてください。

評価方法

ロボットによるピッキングのためにはアイテムの6D poseを推定することが理想です。しかし、本データセットには非剛体のアイテムなど、6D poseの推定が困難な場合があるためバウンディングボックスの重なり率による評価とセグメンテーション領域による評価の2種類の評価方法を採用します。

1. バウンディングボックスの重なり率による評価

バウンディングボックスの重なり率による評価では、検出されたバウンディングボックスと教師信号のバウンディングボックスの重なり率とそのクラスを比較することで評価します。検出されたバウンディングボックス1つに対して、その画像に存在するすべての教師信号のバウンディングボックスと重なり率を比較して、最も重なり率が高いものを推定されたボックスとします。このボックスのクラスと検出されたボックスのクラスが正しければ検出成功とし、異なれば検出失敗とします。

2. セグメンテーション領域による評価

セグメンテーション領域の評価では、距離画像のセグメンテーション結果とラベリングデータを比較することで評価します。セグメンテーション評価のため、1画素単位でセグメンテーション結果とラベリングデータを比較していき、1枚の画像のすべての画素に対して検出率と適合率を算出します。そして、算出された検出率と適合率からF値を求め、F値がしきい値(デフォルトでは0.5)以上であればセグメンテーション成功、しきい値未満ならセグメンテーション失敗と評価します。

Download

2018/07/12 データセットを拡充しました。これに伴い、ファイル構成などを一新しました。

2018/07/18 新規に追加したデータについて不備があったためデータセットを変更しました。

2018/07/20 評価コードをリニューアルしてGitHubで公開しました。また、SegNet・SSD・Multitask DSSDのサンプルコードを公開しました。

画像はすべてbmp形式ですが、サイズが大きいためpng形式に変換したものも用意しました。

- Full Dataset: Team MC²: ARC2017 RGB-D Dataset (tar.gz, 4.72GiB)
- Full Dataset(png): Team MC²: ARC2017 RGB-D Dataset (png) (zip, 3.40GiB)
- アイテムリスト: ItemList.pdf (262KiB)
- セグメンテーションカラーコードリスト: SegmentationTeach.pdf (310KiB) / SegmentationTeach.xlsx (331KiB)
- 評価用ソフトウェア(Bounding Box/Segmentation): MC2ARCdataset_evaluate (GitHub)
- SegNet(ChainerCV)サンプルコード: segnet_arc (GitHub)
- SSD/Multitask DSSD(Chainer)サンプルコード: MT-DSSD (GitHub)

評価用ソフトウェア使用方法

評価方法ごとにプログラムが異なります。また、下記の環境とライブラリが必要となります。

- Python 3.x / 2.x (推奨バージョン $\geq 3.5.1$ or $\geq 2.7.11$)
- OpenCV 3.x / 2.x
- numpy (推奨バージョン ≥ 1.10)
- matplotlib (推奨バージョン ≥ 1.4)

バウンディングボックスの重なり率による評価

データの用意

下記の通りデータを用意してください。

- ./images/ にテスト用画像を入れる
- ./results/ に検出結果(txt形式)を入れる
- ./teach/ にバウンディングボックスの教師ラベル(txt)を入れる

テスト画像およびバウンディングボックスの教師ラベルは、データセットの“test”フォルダ内のものをコピーします。

検出結果は、各値が半角スペースでセパレートされた、教師信号と同じ形式である必要があります。

クラスID Boxの中心のX座標 Boxの中心のY座標 Boxの幅 Boxの高さ

座標と幅と高さは0~1で正規化 (画像のサイズで除算) してください。

プログラムの設定

必要に応じてプログラム内の変数を変更してください。

IMAGE_EXT = “png” : テスト用画像の拡張子を記入。

WAITTIME = 0 : OpenCVのWaitKey()の秒数(ミリ秒)を記入。この値を100程度にしておくと、評価の様子を確認できます。0 ならば確認せず、すぐに評価を行います。

LABEL_FLAG = 0 : 検出結果にクラス名がある場合は1にしてください。

NORMALIZED = 1 : 検出結果が正規化されていない場合は0にしてください。

THRESH = 0.55 : 検出成功と判断する重なり率(IoU)の閾値を記入。

プログラムの実行

“detect_evaluation.py”を実行すると、検出されたバウンディングボックス1つに対して、その画像に存在するすべての教師信号のバウンディングボックスとIoUを比較して、最もIoUが高いものを探索します。見つかったボックスのクラスと検出されたボックスのクラスと2つのボックスのIoUを計算し、テキストファイルに書き出します。

次に、書き出したテキストファイルを読んでConfusion Matrixを計算します。IoUが閾値以上かつクラスが合っていれば検出

成功、そうでなければ検出失敗とします。計算が終了すると、画像形式のConfusion Matrixと、次の3つの値を出力します。評価をやり直す場合は「(result_path)/matchingResults/」と「(result_path)/eval/」をディレクトリごと削除してください。

- ・識別率:検出された物体のうち、クラスが正しかった割合
- ・未検出率:すべての物体のうち、検出できなかった割合
- ・平均IoU:すべてのバウンディングボックスのIoUの平均

セグメンテーション領域による評価

初めに、教師画像とセグメンテーション結果画像をそれぞれ別のディレクトリに用意してください。このとき、同じ画像サイズかつ拡張子をpngにする必要があります。

続いて、“segment_evaluation.py”の次の部分を変更してください。

gt_path = “./label/” : 教師画像のあるパスを記述。

result_path = “./results/” : 結果画像のあるパスを記述。

最後に、“segment_evaluation.py”を実行すると評価を行います。1ピクセルごとに評価をするため時間がかかりますのでご注意ください。

Team MC^2の認識アルゴリズムの評価

上記のデータセットおよび評価用ソフトウェアを用いて、Team MC^2がARC用に開発したアルゴリズムを評価しました。

評価手法

Single-Shot Multibox Detector(SSD): 複数解像度のDefault Boxを回帰することで物体に最もふさわしいBoxを検出するとともに、物体のクラスも識別する。リアルタイムでの一般物体認識が可能。

我々はSSDを独自に実装し、さらに未知物体への対応と既知物体の検出精度向上のためにObjectness(物体らしさ)の識別を同時に行うSSDを開発しました。

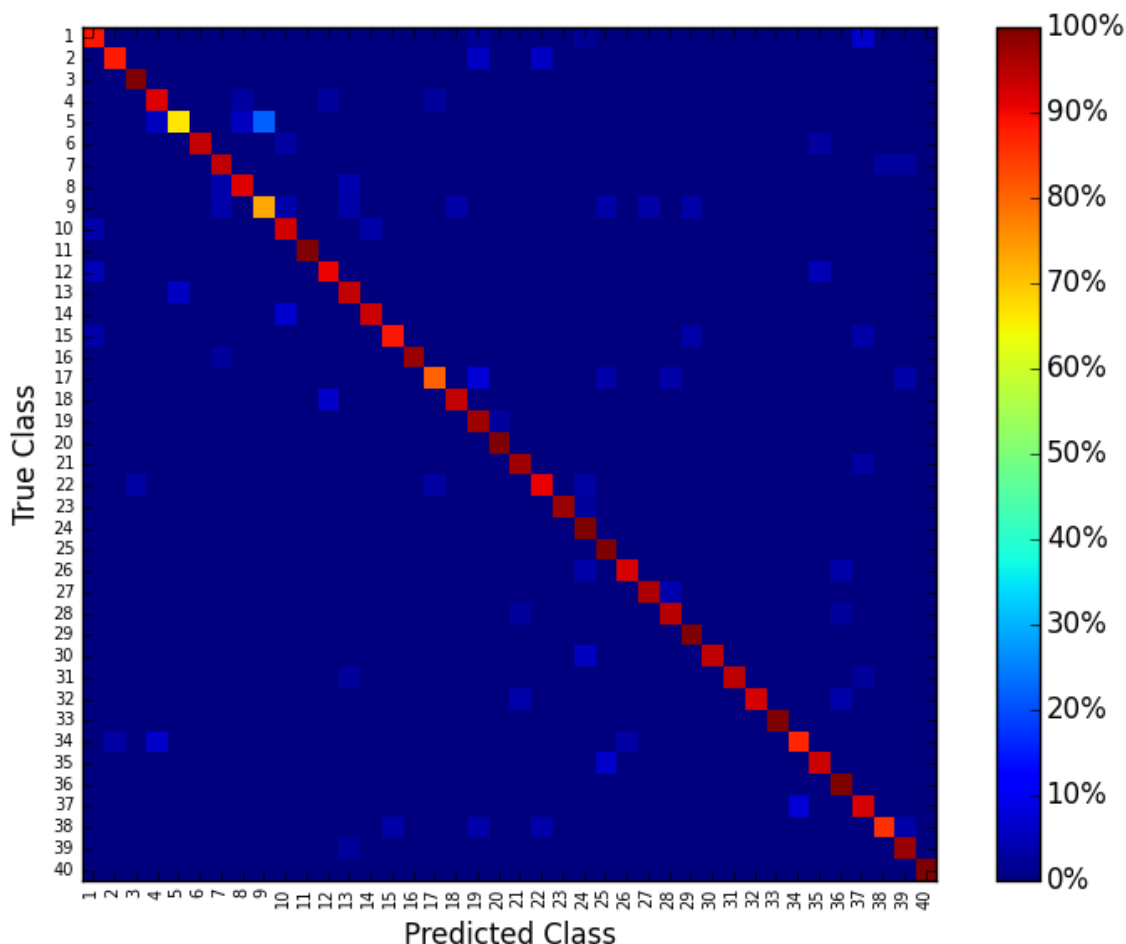
評価結果

識別率: 0.890423572744

未検出率: 0.256164383562

平均IoU:0.792050506884

Confusion Matrix



* 評価結果は随時追加していきたいと考えておりますので、本データセットを用いての評価結果を募集しております。

免責事項

公開しているプログラムやデータに関する情報には十分に注意を払っていますが、その内容について全てを保証するものではありません。プログラム、資料、ウェブサイトの著作者であるMachine Perception and Robotics Groupは、これらの使用ならびに閲覧によって生じたいかなる損害にも責任を負いかねます。プログラムの改良・再頒布によって生じた損害に関しても責任を負いかねます。公開しているプログラムは、上記の免責事項に同意の上、使用者の責任のもとで研究用途に限り、使用して構いません。商用利用される場合については、下記の連絡先にご連絡下さい。

連絡先

藤吉弘亘 : fujiyoshi@isc.chubu.ac.jp

23 May 2018 Deep Learning Dataset 国際会議

ARC2017 RGB-D Dataset for Object Detection and Segmentation

Author R. Araki, T. Yamashita and H. Fujiyoshi

Publication Late Breaking Results Poster on International Conference on Robotics and Automation, 2018

Download: [PDF](#) (English)