

Faster R-CNN を用いた One-click super vision

平野 友基^{1,a)} 新納 浩幸^{2,b)}

概要: 物体検出を行うタスクでは様々な学習モデルが提案されているが、中でも Faster R-CNN は良い性能を示している。オブジェクトクラスを検出するには、通常、バウンディングボックスにより注釈されたオブジェクトを持つ大規模なセットが必要である。しかし、手作業でバウンディングボックスを描くのは非常に大変な作業である。そこで中心をクリックすることでアノテーションの時間を大幅に短縮する click supervision という手法が提案された。本稿では、detectron2 を用いた Faster R-CNN による one-click supervision の提案。これを風船の物体検出を学習するカスタムデータセットを用いて行い、one-click supervision を用いて学習したモデルとカスタムデータセットのみを使用して学習したモデルの精度を比較した。この結果、モデルの精度はカスタムデータセットを用いた場合には及ばないが、学習の効果は見られた。これに基づき、提案手法では精度の向上は見られるが、通常の方法でアノテーションされたデータに精度を近づけるには改善が必要であると結論付けた。

One-click supervision using Faster R-CNN

Abstract: Various learning models have been proposed for the task of object detection, among which Faster R-CNN has shown good performance. To detect a class of objects, we usually need a large set of objects annotated with bounding boxes. However, drawing bounding boxes by hand is a very difficult task. To solve this problem, a method called "click supervision" has been proposed, which greatly reduces the annotation time by clicking on the center. In this paper, we propose a one-click supervision method based on Faster R-CNN using detectron2. This was done using a custom dataset for learning balloon object detection, and the accuracy of the model trained using one-click supervision was compared with the model trained using only the custom dataset. We compared the accuracy of the model trained with one-click supervision with that of the model trained using only the custom dataset. As a result, the accuracy of the model was not as good as that of the model trained using the custom dataset, but the learning effect was observed. Based on this, we conclude that the proposed method improves the accuracy, but needs improvement to bring the accuracy closer to that of the data annotated in the usual way.

1. はじめに

物体検出は入力された画像から物体の位置とカテゴリーを検出するタスクである。物体検出はオブジェクトクラス検出器によって行われ、これを訓練することで、物体検出の精度を高める。物体検出モデルには様々な物が提案されている [1][2]。中でも Faster R-CNN [3] は優れた性能を示している。しかし、物体検出において、オブジェクトクラ

ス検出器を訓練するためには、画像にバウンディングボックスと呼ばれる対象を囲む矩形の情報を与える必要があり、これを一からすべて手作業で行うには時間がかかる。そこで物体の中心をクリックするのみで行うことのできる click supervision [4] という手法が提案された。click supervision とは、物体の中心点をクリックすることで、画像にアノテーションデータを与えるもので、通常のアノテーションデータの作成の場合と同等の精度で9から18倍速く作成することができる。

ただし click supervision のベースとなる物体検出アルゴリズムには Fast R-CNN が用いられている。ここではこの部分を Faster R-CNN に変更することを試みる。またプログラム内で用いるライブラリは Detectron2 のものを利用した。Detectron2 とは、Facebook AI が開発した、

¹ 茨城大学大学院理工学研究科情報工学専攻
Graduate School of Science and Engineering, Ibaraki University, Department of Computer Science and Engineering

² 茨城大学大学院理工学研究科情報科学領域
Graduate School of Science and Engineering, Ibaraki University, Department of Computer Science and Engineering

a) 21nm750l@vc.ibaraki.ac.jp

b) hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

PyTorch ベースの物体検出のライブラリである。これは、他のライブラリと比較して少ないコードで Bounding box や Instance Segmentation 等を生成する物体検出を実装することができる。

本論文では上記のように実装した click supervision の改良版を One-click supervision と名付けて提案する。また Detectron2 内の風船の物体検出を学習するカスタムデータセットを用いて提案手法により学習したモデルの精度を調べる。

2. One-click supervision

2.1 click-supervision

Click-supervision [4] では、中心をクリックすることでアノテーションを行う。これは、物体の中心点 (center-click annotation) に MIL(Multiple Instance Learning)[6] を適用することでバウンディングボックスを定位する手法で、これによってアノテーション時間を大幅に短縮できる。論文 [4] の実験では、click-supervision はわずかなアノテーションの追加作業で弱教師化技術で生成された検出器よりも優れた性能を発揮し、手動で描画されたバウンディングボックスから学習された検出器に近い学習結果を示した。

2.1.1 click-supervision の流れ

click-supervision の workflow を図 1 をもとに説明する。instruction で click annotation を行い、次に Annotator training を行う、これは図 2 に示されているような画像に対して、黒い部分の面積の中心をアノテーションしてもらい、実際との中心とのずれのフィードバックを受け取る。これを中心との誤差が 20px 以下になるまで繰り返す。ここで求めた中心との誤差の標準偏差 σ_{bc} を実際の作業時に使用する。

このテストをクリアしたら、実際の作業へと移行する。Annotating images では、特定の枚数を単位として、アノテーターに物体の中心のクリックをしてもらう。ここでは、Quality を担保するために、もともと Ground Truth を持っているデータをバッチごとにランダムに混ぜて精度計測を行う。この時、精度が一定に満たなかった人のデータは受け付けない

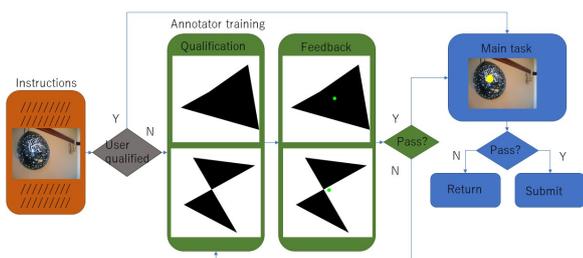


図 1 click annotation をクラウドソーシングで行う Workflow

しかし、論文 [4] における物体検出の学習モデルには Fast R-CNN が用いられていた。一般的に Faster R-CNN の方が学習が素早く精度も高いとされているため、One-click supervision を Faster R-CNN を導入したときの学習の様子を見た。

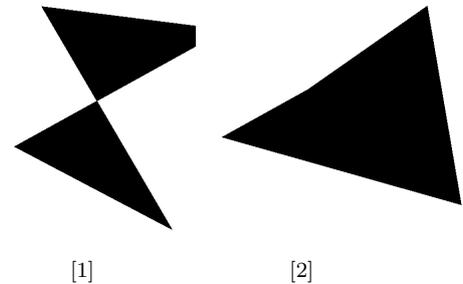


図 2 annotator training 用に生成された画像

そのため、Pre-train された AlexNet CNN と SVM を使って下記の 2 ステップを交互に回す。

- re-localization 識別機 A を使い物体候補を探す。この時、物体候補 p のスコアは識別機 A と Objectness[2] を用いた物体候補らしさ O を使い式 (1) に示す計算で求める。 p は提案された領域である。

$$S_{ap}(p) = \frac{1}{2} \cdot A(p) + \frac{1}{2} \cdot O(p) \quad (1)$$

また、クリックに最も近いオブジェクトを選択するだけでは正確なバウンディングボックスは得られない。そこで、クリックした点 c と annotator training 時に求めた σ_{bc} を用いて、スコア関数 S_{bc} によって中心尤度を求める。これを式 (2) に示す。 p は提案された領域、 c_p はその中心点、 c はクリックした点である。 σ_{bc} は c_p が c から離れるにつれて値が小さくなるよう制御するものである。

$$S_{bc}(p; c, \sigma_{bc}) = e^{-\frac{\|c_p - c\|^2}{2c^2 \sigma_{bc}^2}} \quad (2)$$

S_{ap} と S_{bc} を積としてバウンディングボックスを定位する。これを式 (3) に示す。

$$S_{ap}(p) \cdot S_{bc}(p; c, \sigma_{bc}) \quad (3)$$

- re-training 提案された領域の中で (3) で求めたスコアの最も大きいものを Positive として、識別機 A を SVM にて学習させる。

一定回数イテレーションを回した後、re-training で識別機 A を Fast R-CNN にして再学習する。

2.1.2 Faster R-CNN

Faster R-CNN は物体検出モデルの一つで、R-CNN, Fast R-CNN から派生し、より高速で高精度になったものである。Fast R-CNN までの領域提案には従来技術である Selective Search を使用しておりこの部分の処理が全体の内の多くの時間を占めていた。具体的には、Fast R-CNN では一枚の画像の処理に 2.3 秒かかるが、そのうち 86% が Region Proposal に費やされ、ボトルネックとなっていた。検出までの流れを図 3 に表す。

そこで、Faster R-CNN は Region Proposal も Region Proposal Network というニューラルネットワークに置き換えて物体検出モデルを全て DNN にし、End-to-End で学習することによって高速化を実現した。検出までの流れを図 4 に表す。

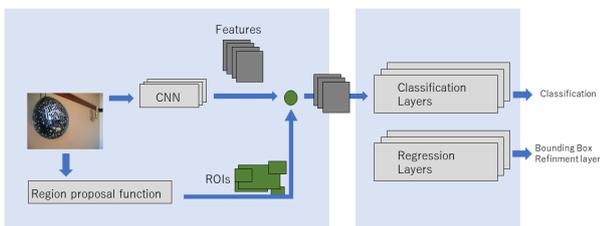


図 3 Fast R-CNN の検出までの流れ

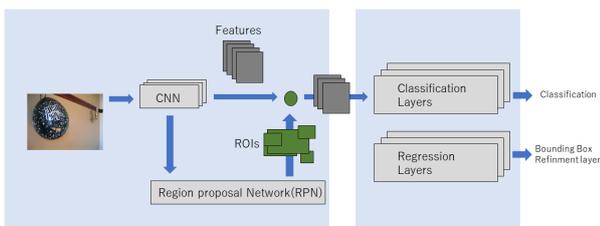


図 4 Faster R-CNN の検出までの流れ

2.1.3 実装方法

[4] では学習モデルに Fast R-CNN が使用されており、領域の提案と決定を行う識別機 A と別々に学習を行っていた。本研究では Faster R-CNN のスコアと RPN による領域提案を利用することで Faster R-CNN のみでの学習でこれを行った。

3. 実験

3.1 実験方法

Faster-rcnn を用いた One-click supervision を用いて、

COCO データセットで学習済みのモデルを使い Balloon segmentation dataset の転移学習を行う。まず、annotator のトレーニングを行う。トレーニングは、図 5 に示すような生成された画像の中心をクリックすることで行う。この作業を 20 回行い誤差の平均が 20px 以下になったらクリアとする。この時、クリックした点と中心点の誤差の標準偏差を求める。次に、表示されたデータセット中の画像の物体の中心をクリックし、クリックデータを収集する。これをもとに、COCO の学習済みの Faster R-CNN のモデルを click annotation された balloon dataset によるモデルの学習と評価を行う。

学習は、Faster-rcnn の RPN によって提案された領域のスコア $S_{ap}(p)$ と式 (2) によって求めたスコアの積式 (3) の最大値を Positive として、Faster-rcnn の精度を上げることで行う。

train 用の画像は balloon segmentation dataset 内の train 用の画像 61 枚にそれぞれクリックデータをもとにしたバウンディングボックスの定位を行い、学習用のデータセットとする。これを 1 セットとし、エポック数は 300 とした。また、このプログラムの実装は Detectron2 のチュートリアルとして公開されている Detectron2 Beginner's Tutorial を参考にした。また、データセットのアノテーションデータをもとに学習を行うプログラムで学習したモデルの評価を行う。

これらの手順をもとに、one-click supervision の精度の向上も試みた、方法としては、300 エポックごとにクリックデータと学習モデルをもとに再びバウンディングボックスの定位を行うものと、定位を決定する式を (4) とし、スコアを二乗することでその重要度を高め、モデルの精度を見た。

$$S_{ap}(p)^2 \cdot S_{bc}(p; c, \sigma_{bc}) \quad (4)$$

3.2 実験結果

annotator training の実験結果を図 5 と表 1 に示す。画像内の緑色の点が黒い図形の中心点で青い点が実際にクリックした点である。また、表 1 は誤差とその平均値、標準偏差である。

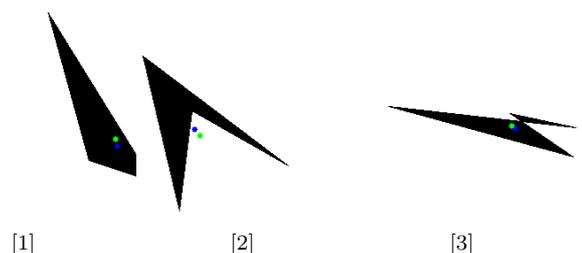


図 5 annotator training の実行例

表 1 誤差とその標準偏差

誤差	8.004	9.953
	9.906	5.506
	4.6502	5.315
	8.721	32.311
	4.123	5.534
	6.374	14.502
	14.752	23.856
	15.988	8.515
	12.816	5.408
	6.973	4.472
誤差の平均値	10.384	
誤差の標準偏差	12.524	

これより、誤差の標準偏差は $\sigma_{bc} = 12.524$ となる。これを本研究では、式 (2) 内の σ_{bc} に使用した。今回のアノテーターは私のみなので、balloon データセットの学習に使うコードでは変数を直接書き換えた。

次に実際に使用するデータセットによる one-click supervision を行った。実行結果を図 2 に示す。

青い点がクリックした点で、緑のボックスがクリックデータをもとに生成されたバウンディングボックスを再現したものである。比較的正確にアノテーションデータを作成できた画像もあるが、図 6 内の画像、annotated3.eps や annotated4.eps ではクリックした点から大きく離れた位置にバウンディングボックスが生成された。



[1] annotated1.eps [2] annotated2.eps



[3] annotated3.eps [4] annotated4.eps

図 6 One-click supervision の実行例

このクリックデータによりアノテーションされた Balloon segmentation dataset をもとに、COCO データセットにより学習済みの Faster R-CNN の転移学習を行った。結果を表 2 に示す。また、データセット内のアノテーションデータを元に行った学習結果を表 3 に示す。これより、データセット内のアノテーションデータには及ばないが、one-click supervision による学習の成果が見られることが分かった。また、式 (3) を式 (4) に変更することでスコ

表 2 one-click supervision で生成された画像での学習

イテレーション数	AP	AP50	AP75	APs	APm	APl
300	62.5	79.9	78.9	1.1	46.2	79.0
600	56.5	76.3	72.4	2.8	42.4	73.0
900	69.0	83.3	77.7	5.3	53.7	84.4
1200	71.4	85.4	82.5	5.1	57.5	85.9
1500	69.9	85.4	84.0	6.8	56.3	83.6
1800	72.2	85.4	82.8	4.5	58.1	86.5
2100	71.3	85.5	82.7	4.5	59.7	84.6
2400	72.0	85.2	82.4	5.0	58.3	86.5
2700	71.7	84.9	82.1	5.0	57.0	86.8
3000	70.6	85.5	82.6	12.3	56.4	84.7

表 3 データセットの bounding box を用いた学習

イテレーション数	AP	AP50	AP75	APs	APm	APl
300	62.1	84.4	73.1	10.6	54.9	73.5
600	77.5	90.6	85.8	17.2	61.7	90.4
900	77.4	91.4	90.6	18.6	63.7	88.6
1200	78.7	91.5	88.1	33.4	64.1	90.5
1500	80.7	92.3	88.3	34.0	65.1	92.5
1800	78.6	91.4	87.3	27.9	62.0	92.1
2100	79.2	91.4	87.2	27.7	63.3	92.6
2400	80.3	91.4	87.1	27.9	64.2	93.4
2700	78.9	91.5	87.0	28.0	64.7	91.6
3000	79.7	91.5	87.1	31.4	67.3	91.6

表 4 $S_{ap}(p)$ を二乗したときの学習

イテレーション数	AP	AP50	AP75	APs	APm	APl
300	66.4	83.4	76.8	3.6	52.9	82.5
600	60.7	82.1	70.7	3.5	53.6	73.5
900	69.9	84.2	81.8	4.2	53.9	85.2
1200	70.7	84.7	81.5	3.0	53.4	86.2
1500	70.4	84.3	83.1	3.4	54.8	85.4
1800	68.7	83.7	82.6	2.5	51.9	84.0
2100	69.5	83.5	82.8	2.5	52.9	85.1
2400	71.3	83.7	82.9	3.4	52.8	88.0
2700	71.0	85.3	84.5	14.4	53.2	85.6
3000	69.4	84.3	83.5	14.4	52.0	84.3

表 5 one-click supervision による学習済みモデルでバウンディングボックスの再定位

イテレーション数	AP	AP50	AP75	APs	APm	APl
学習済みモデル	70.6	85.5	82.6	12.3	56.4	84.7
300	45.1	68.0	49.6	11.4	55.6	58.0
600	16.1	51.9	1.4	2.2	16.0	21.1
900	15.6	46.9	3.8	0.5	12.4	21.5

表 6 データセットによる学習済みモデルでバウンディングボックスの再定位

イテレーション数	AP	AP50	AP75	APs	APm	APl
学習済みモデル	80.5	92.4	86.6	19.1	66.3	92.8
300	67.5	88.1	76.9	24.4	55.5	80.5
600	35.2	64.6	33.0	6.0	30.3	43.4
900	30.8	63.1	11.0	1.2	23.0	38.3

アによる値の変化を大きくしたときの結果を表 4 に示す。

スコア関数を式 (4) に変更しても結果に大きな変化は見られなかった。

次に, one-click supervision によって一定回数学習を行った後, その学習済みモデルを元に再びバウンディングボックスの定位を行った。バウンディングボックスが再定位された画像を図 7 結果を表 5 に示す。すると, 物体検出の精度は大幅に低下した。データセット内のアノテーションをもとに同じ手順を繰り返したが, これも同様に物体検出の精度は大幅に低下した。結果を表 6 に示す。



[1] annotated1.eps [2] annotated2.eps

図 7 学習済みモデルとクリックデータによる bounding box の再定位

4. 考察

クリックデータをもとにした学習では, one-click supervision を Detectron2 を用いて実装した。結果はデータセットにあった教師データほどではないが学習の成果が見られたため, Faster R-CNN による one-click supervision は有効であるといえる。

また, one-click supervision によるバウンディングボックスの定位がうまくいかなかった図 6 内の画像 annotated3.eps や annotated4.eps のようなものは生成されるバウンディングボックスが一箇所に集中する傾向があった。

例えば, annotated4.eps では上側 3 つの風船をクリックした位置から大きくはずれて右下の風船にバウンディングボックスが集中している。これより, この大きく離れた位置に定位するボックスは物体尤度のスコアの値 (S_{ap}) が大きく, 式 (3) $S_{ap} \cdot S_{bc}$ において, S_{bc} の値の影響が小さくなったからだと考えた。ただ, このような精度の高いボックスはある風船のボックスとしては良い結果を示していた。これより, モデルの精度を上げるために, スコア関数の見直しを考えた。結果として, S_{ap} を二乗することで, 領域候補の尤度の優先度を下げつつ尤度の差によるスコアの開きを大きくするといったような式を調整する方法をいくつか試した。

しかし, $S_{ap}/2$ をスコアとする, σ_{bc} を 2 倍するといった式の調整を行ったが物体検出の精度に変化は見られなかった。これより, 式の変更により one-click supervision の精度向上を試みるには何か新しい指標を取り入れるか別の視点での改善が必要であると考えられる。

5. おわりに

本論文では, Detectron2 と Faster R-CNN を用いた One-click supervision による学習を行った。またバウンディングボックスの定位に工夫を施し, 精度の向上を試みた。結果として, Faster R-CNN を用いた One-click supervision による学習の成果は見られたが, 精度の向上のために行った工夫では成果が見られなかった。精度の向上のためには, 別の視点からのアプローチが必要である。

参考文献

- [1] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, UC Berkeley: Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5), Conference on Computer Vision and Pattern Recognition (CVPR), (2014.10.22).
- [2] Ross Girshick, Microsoft Research: Fast R-CNN, International Conference of Computer Vision (ICCV), (2015).
- [3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Neural Information Processing Systems 28 (NIPS), (2015).
- [4] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, Vittorio Ferrari, University of Edinburgh, Google Research: Training object class detectors with click supervision, Conference on Computer Vision and Pattern Recognition (CVPR), (2017).
- [5] Bogdan Alexe, Thomas Deselaers, Vittorio Ferrari Computer Vision Laboratory, ETH Zurich: What is an object?, Conference on Computer Vision and Pattern Recognition (CVPR), (2010).
- [6] Boris Babenko, Ming-Hsuan Yang, Serge Belongie: Visual Tracking with Online Multiple Instance Learning, Conference on Computer Vision and Pattern Recognition (CVPR), (2009).