

物体検出を用いたキーポイント推定手法の提案と直方体の頂点検出の実装

平野 龍一[†] 岡崎 勝守[†] 福原 義久^{††}

[†] 武蔵野大学データサイエンス学部 〒135-8181 東京都江東区有明 3-3-3

^{††} 武蔵野大学 アジア AI 研究所 〒135-8181 東京都江東区有明 3-3-3

E-mail: [†] {s2122068, s2122013}@stu.musashino-u.ac.jp, ^{††} {y-fuku}@musashino-u.ac.jp

あらまし 本稿では、物体検出手法を活用したキーポイント推定手法、及び提案手法を用いた直方体の頂点検出について述べる。運送用に利用される外装箱は直方体であることが多く、この直方体の頂点を正確に検出し寸法や体積等を推定することが物流やロボティクスの分野において求められている。しかし従来の頂点検出アルゴリズムでは、直方体状の物体の頂点以外の特徴点を誤検出してしまうことも多く、必要な頂点のみを検出することが困難である。本手法では頂点を一つの物体として見立て深層学習を用いた物体検出モデルに学習させることで、直方体の頂点の正確な検出に成功した。

キーワード 画像認識, パターン認識, ユーザ支援

1. はじめに

近年、Electric Commerce（以下 EC）の市場規模が拡大している。特に、物販系分野における EC 市場規模は新型コロナウイルスの感染拡大に伴う外出自粛の呼びかけにより大幅に拡大した。令和 3 年度電子商取引に関する市場調査報告書[1]によると、国内の商取引のうち EC が閉める割合を示す EC 化率について、B to C においては 2021 年現在で 8.78% となっている。これは、2013 年の 3.85% と比較すると 4.93% 増加している。また、B to B においては 2021 年現在で 35.6% となっている。これは、2017 年の 29.4% と比較すると 6.2% 増加している。このように EC 利用者の増加、及び市場規模の拡大に伴い、取り扱われる物品、商品も増大している。結果、保管や運送のためのスペースの効率的な利用が物流において求められている。また、物品管理のためにロボットが活用される場面も今後さらに増加するであろう。

スペースの効率的な利用やロボットの活用のためには、物体の形状を正確に把握する必要がある。現在、運送用の外装箱の形状としては直方体が多く利用されており、この寸法や体積を正確に推定することは上記物品管理・運送に有用であると考えた。

直方体の寸法及び体積を推定するためには、直方体の各頂点を検出する必要がある。しかし、従来の頂点検出アルゴリズムでは直方体の頂点以外の特徴点を誤検出してしまうことが多く、寸法及び体積の推定に必要な頂点のみを検出することが困難であるという課題がある。

本稿では、上記課題を解決し、正確に直方体の頂点のみを検出するための、物体検出を用いたキーポイント推定手法の提案とその実装について示す。

提案手法では、直方体の頂点一つ一つを、それぞれ一つの物体として見立て、深層学習を用いた物体検出モデルに学習させることで、各頂点の正確な検出に成功した。

本稿は、次のように構成される。2 節では、既存の頂点検出アルゴリズムを含めた関連研究について示す。3 節では、物体検出を用いたキーポイント推定手法について示す。また、提案手法を用いた直方体の頂点検出モデルについても示す。4 節では、3 節にて実装した直方体の頂点検出モデルの検証について示し、5 節ではこのモデルのさらなる精度向上の可能性を目的とする実験について示す。最後に、6 節では本稿のまとめとして、結論及び今後の展望を示す。

2. 関連研究

本節では、提案手法の関連研究について述べる。

本研究では、直方体の各頂点を求めるが、これは面を構成する四角形の頂点を検出できれば良いともいえる。多種多様な特徴点の検出ではなく、四角形の頂点のみの検出に関する研究について、斉藤らは、エッジ画素検出と組合せ Hough 変換を用いて長方形の頂点検出を実現している[2]。この研究では、検出対象を長方形に限定することで、Hough 変換を使用した処理を簡素化し、検出の信頼性を向上させている。

趙らは、物体検出を用いた頂点検出によるカード帳票の検出について述べている[3]。この研究では、アノテーションを行う際に、検出対象を示すバウンディングボックスの中心点がカード帳票の各頂点と重なるようにアノテーションをおこなっている。

提案手法では、バウンディングボックスの中心点がより正確に対象物体の頂点を捉えることができるよう、キーポイント形式でのアノテーションを行った。また、趙らの研究では、カード帳票を対象としているため、対象物体と背景の境界となる頂点を検出対象としているが、本実装では、直方体の頂点検出という目的から、対象物体領域内の頂点検出にも取り組んだ。

3. 物体検出を用いたキーポイント推定手法を利用した直方体の頂点検出

3.1. 提案手法の流れ

本節では、提案手法の流れについて述べる。提案手法は図1に示したとおり、直方体画像データセットに対するキーポイント形式でのアノテーション、直方体頂点検出モデル作成のための yolov7 pytorch txt 形式への変換、モデル作成という流れから成る

3.2. 直方体画像データセットについて

本研究では、独自に撮影した段ボールの画像 300 枚を直方体画像データセットとして用意した。撮影に用いた機材に関する情報を表1に示す。また、撮影した画像の一例を図2に示す。図2に示す画像と同様に、7つの頂点が明確に視認できる画像のみを使用した。今回は約 330×250×175mm(幅×奥行×高さ)から成る運送用外装箱一種類を上記条件に沿って様々な角度から撮影し、データセットを構成した。

3.3. 箱の頂点に対するキーポイント形式のアノテーション

提案手法では頂点を得るためバウンディングボックスを用いているが、正確に頂点位置をアノテーションするためにキーポイント形式を用いたアノテーションを最初におこなう。なお、キーポイント形式でのアノテーションには labelme[4]を使用した。

直方体を撮像した場合、画像内には最大7つの頂点が確認できる。これらの7つの頂点に対して、それぞれ図3の通りにラベル付けを行った。7つの頂点に対してそれぞれ固有のラベル付けを行うことで、作成した頂点検出モデルを使用して、各頂点を検出した際に位置関係も同時推定することが可能となり、より正確

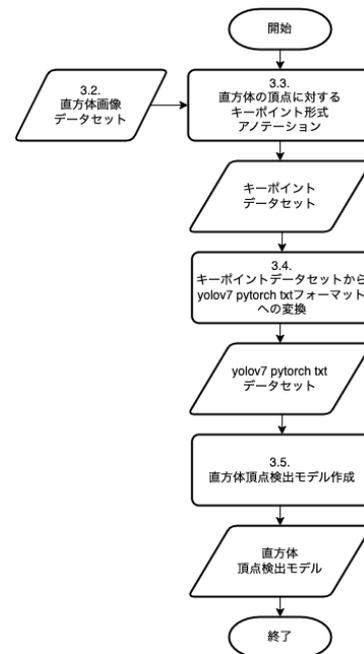


図1 提案手法の流れ

表1 直方体画像データ撮影機材情報

使用機種	iPhone 13 128GB
使用アプリ	標準カメラ
解像度	1440 × 1440



図2 撮影した画像の一例

な、直方体の領域推定と辺の描写等が可能となる。出力ファイルは、画像とそれに対応するラベル情報およびキーポイント座標を含む LabelMe Json 形式[5]の json ファイルとなる。この形式のラベル情報を対応する画像に適用した場合、図3の通りになる。また、このファイル群をキーポイントデータセットと呼ぶ。

3.4. キーポイント形式から YOLOv7 PyTorch TXT 形式への変換

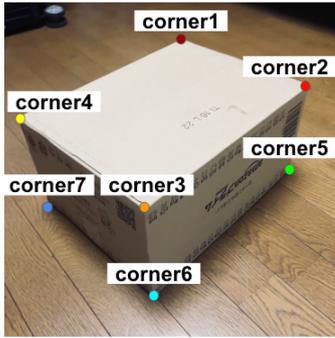


図3 各頂点のラベル付け
及びキーポイントアノテーション



図4 YOLOv7 PyTorch TXT 形式の適用

本節では、3.3.節にて生成されたキーポイントデータセットを物体検出モデルに適用するための、バウンディングボックス形式への変換について述べる。このバウンディングボックス形式は、YOLOv7 PyTorch TXT 形式[6]であり、YOLOv7 を使用したモデル作成のための形式である。変換の方法としては、3.3.節でラベル付けされた各頂点の座標から上下左右同じピクセル分拡張し、正確に頂点を中心とした正方形のバウンディングボックスを設定する。拡張するピクセル分は任意に設定する。本実験では、1440×1440 ピクセルの画像内で、キーポイントを中心に上下左右 20 ピクセルずつ拡張し、バウンディングボックスの一边がそれぞれ 40 ピクセルとなるように変換した。この変換を通して、各画像とその画像に対応したバウンディングボックス情報を含む TXT ファイルから成る YOLOv7 PyTorch TXT 形式のデータセットを生成する。この形式のラベル情報を対応する画像に適用した場合は、図 4 の通りになる。また、このプロセスの最後で、画像を YOLOv7 の入力層サイズである 640x640 ピクセルにリサイズした。

3.5. 直方体の頂点検出モデル

本節では、3.4.節で生成されたバウンディングボックスデータセットを使用した直方体頂点検出モデル作成について述べる。

データセットには 300 枚の外装箱画像を用いた（表

表2 データセットの分割割合とその枚数

用途	分割割合(%)	枚数(枚)
学習用	70	210
検証用	20	60
テスト用	10	30

2). 物体検出モデルとしては YOLOv7[7], R-CNN[8], Faster R-CNN[8], SSD[9]などが挙げられるが, Ahmed の検証[10]によると, YOLOv7 の以前のモデルである YOLOv5[11]が Faster R-CNN より精度, 速度の面で優れていることが示されている。また, Zhu と Yan によって SSD と YOLOv5 の比較実験[12]がされており, YOLOv5 が精度, 認識速度の面で優れていることが示されている。YOLOv7 は YOLOv5 と比較した場合, この二つの点で YOLOv5 より優れていることが示されている[7]. よって, 提案手法では, YOLOv7 を選定した。今回, 学習時のバッチサイズは 8 とした。

この学習における, box loss, objectness loss, classification loss, 及び mAP の結果を図 5 に示す。図 5 では, 後述の mAP の観点から有効であると考えられる 55 エポックまでを示した。

Box loss はアルゴリズムが対象オブジェクトの中心を捉えているか, そして, 提案されたバウンディングボックスがどれだけ対象オブジェクトをカバーできているかに関する損失を示す。Objectness loss は提案された領域内に対象オブジェクトが存在するかどうかに関する損失を示す。Classification loss は捉えた対象オブジェクトに対するラベル付けが適切かどうかに関する損失を示す。そして, mAP は予測されたバウンディングボックスのうち, 重なる領域の比率を表す評価指標である IoU が, 0.5 以上で正しくオブジェクト検出できている度合いを示す precision, そして, 対象となるオブジェクトのうち, 同じく, IoU が 0.5 以上でアルゴリズムにより検出された度合いを示す recall を合

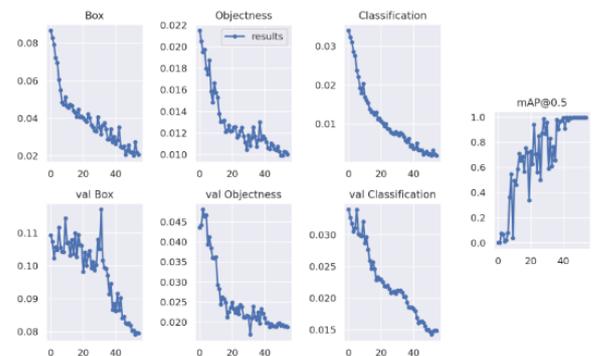


図5 頂点検出モデル作成時の
box loss, objectness loss, classification loss
及び mAP

わせた指標である, AP を全てのクラスに対して平均したものである. 図 5 に示す通り, box loss, objectness loss, classification loss において, 学習用データ及び検証用データの両方に対して, 確実に減っており, 過学習も少なくモデルを作成できていることがわかる. また, mAP に関して高い値を示していることがわかる.

4. モデルの検証

本節では, データセットのうち, テストデータに対する実験の結果を示す.

4.1.1. 実験内容

テストデータである 30 枚の画像を, 3 節で作成したモデルに入力し, 出力された推定頂点座標とそのラベルを, アノテーションされたデータを比較する. 検出された頂点のうち, 正しい頂点であると推定する閾値としての信頼度は 0.5 とした. 検出した項目のうち, 推定頂点座標と実測頂点座標との距離を絶対誤差とする. また, 3.4 節で述べた通り, 画像のサイズは 640×640 ピクセルであるため, この画像内で取れる最大距離は約 905.10 ピクセル(有効数字 3 桁)である. これを誤差 100%として, 絶対誤差の値を除算した値を相対誤差として表す.

4.1.2. 検証結果

誤差最小値, 誤差中央値, 誤差最大値, 誤差平均値, それぞれの絶対誤差, 相対誤差を有効数字 3 桁で表 3 に示す. 誤差最小値は絶対誤差基準で 0 となっており, 完全に一致していることがわかる. また, 誤差最大値に関しても相対誤差基準で 0.6%弱となっている. 相対誤差の中央値と平均値に関しては差分が 0.001%となっており正規分布に近い分布であることがわかる.

5. バウンディングボックスの大きさによる精度比較

3 節の実装では, 3.4 節のキーポイント形式から YOLOv7 PyTorch TXT 形式への変換において, リサイズ前の画像に対するバウンディングボックスの大きさを, 一辺 40 ピクセルと任意に定めた. しかし, このバウンディングボックスの大きさによって, 精度に影響を及ぼす可能性を考慮し, 本節では, バウンディングボックスの最適な大きさを求めるために, 大きさごとの比較を行った.

比較には, 4 節で使用したテストデータと同じ 30 枚のデータを使用した. サイズとしては, 一辺を 20 ピクセルから 200 ピクセルまで, 20 ピクセル刻みで 10 個のデータセットを生成し検証した.

表3 テストデータに対する絶対誤差及び相対誤差

代表値	絶対誤差(pixel)	相対誤差(%)
誤差最小値	0.00	0.00
誤差中央値	1.50	0.166
誤差最大値	5.41	0.598
誤差平均値	1.51	0.167

5.1. 誤差比較

まずは, 4.1.1 節で述べた実験方式で誤差を算出した. サイズによって誤検出や重複検出によって, 外れ値が生じる可能性があるため, 比較項目として四分位範囲を追加した. サイズごとの比較項目の推移を図 6 に示す. グラフから, 一辺の長さが 20 ピクセルから 140 ピクセルまでは各比較項目に大きな変化がないことがわかる.

これより, 誤差基準では, 適切なバウンディングボックスサイズは 20 ピクセルから 140 ピクセルの間であることがわかる.

5.2. 未検出, 重複検出数比較

バウンディングボックスのサイズによっては, 検出されない頂点や, 重複して検出される頂点が存在する可能性がある. そのため, 本節では, この検出された頂点ラベル数, 及び重複検出された頂点の数を比較する. 比較項目としては, 検出された頂点ラベル数に関する最小値, 中央値, 最大値, 平均値, 最頻値の代表値と重複検出数の 6 項目である. この 6 項目のバウンディングボックスごとの推移を図 7 に示す.

グラフから, 検出された頂点ラベル数に関しては, 一辺の長さが 40 ピクセルから 140 ピクセルまでは, どの値も同じく 7 を示していることがわかる. このことから, 上記範囲内では 7 つ全てのラベルが検出されていることがわかる.

重複検出数に関しては, 一辺の長さが 140 ピクセルまでは存在せず, 180 ピクセルから増加していること

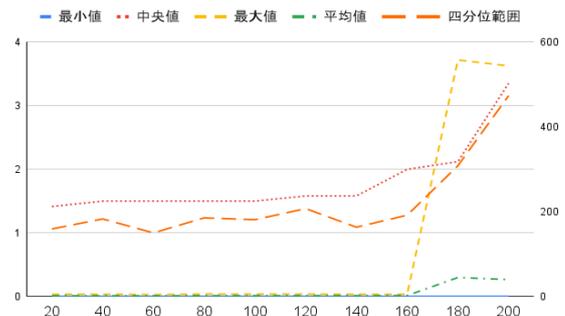


図6 バウンディングボックスサイズごとの誤差比較

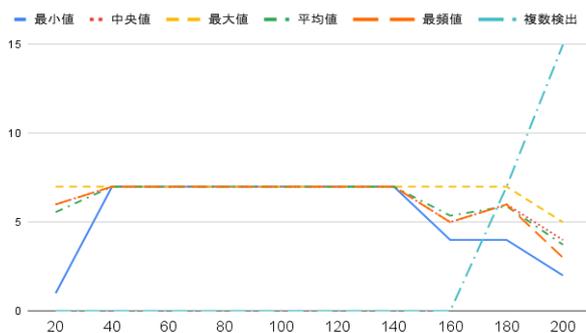


図7 バウンディングボックスサイズごとの未検出、重複検出数比較

がわかる。

これより、未検出、重複検出数基準では、適切なバウンディングボックスサイズは 40 ピクセルから 140 ピクセルの間であることがわかる。

5.3. バウンディングボックスの最適なサイズについて

4.1.及び4.2.の結果をまとめると、最適なバウンディングボックスのサイズは一辺 40 ピクセルから 140 ピクセルの間であることがわかる。よって、3 節で設定した一辺 40 ピクセルのバウンディングボックスは適切であったと言える。

6. 結論

本稿では、正確に直方体の頂点のみを検出するための、物体検出を用いたキーポイント推定手法の提案とその実装について示した。

提案手法では、アノテーションにおいて、より正確に、頂点を中心としたバウンディングボックスを設定するため、キーポイントアノテーション手法を用いた。また、検出する際には、バウンディングボックスの中心となる点を頂点とし、各頂点を別々のラベルで認識することによって、直方体の領域のより正確な推定と辺の描写を可能としている。

この、任意の特徴点一つ一つを別のラベルで認識するように物体検出モデルに学習させることは、二次元形状のみならず、三次元形状の物体を対象とする場合における頂点検出も可能としており、他のキーポイント推定モデルの作成にも応用が可能であると期待される。

また、今回の作成された実験モデルに対する、5. 節での検証を通しては、バウンディングボックスを適切なサイズに頂点を中心として設定することで、誤差を抑えることに成功した。

7. 今後の展望

今後の課題としては、まずは、検出する運送用外装箱の多様化が挙げられる。今回は一種類の寸法の直方体を使用した。様々な寸法の直方体を学習させ、新たな検出モデルを作成することが、実用化に向けて必要と考えられる。また、撮影された外装箱の角度によっては、確認できる頂点の数が少ないケースも考えられ、そのような場合にも使用できるように改良することも挙げられる。

また5 節の比較実験により、設定するバウンディングボックスのサイズによってその精度に影響を及ぼすことがわかったが、実際の利用状況下では、カメラから物体までの距離や箱の大きさ、画素数などの影響が考えられ、状況に応じた適切なバウンディングボックスサイズが必要となる可能性も考えられる。

最後に、各頂点の正確な位置関係を利用した、寸法及び体積の推定にも挑戦したい。対象物体までの距離を推定するなどして、この課題を解決することで、物流、ロボティクスにおける活用幅が広がると考えられる。

参考文献

- [1] 経済産業省 商務情報政策局 情報経済課, 令和3年度 電子商取引に関する市場調査報告書, <https://www.meti.go.jp/press/2022/08/20220812005/20220812005-h.pdf>
- [2] 齊藤文彦, 島知也, “組合せ Hough 変換による画像内長方形物体検出”, 精密工学会誌 Vol.71, No.1, 2005
- [3] 趙振, 橋岡佳輝, “OCR のための物体検出を用いたカード帳票検出手法”, The 36th Annual Conference of the Japanese Society for Artificial Intelligence, 2022
- [4] Labelme, <https://github.com/wkentaro/labelme>
- [5] Roboflow, What is LabelMe JSON?, <https://roboflow.com/formats/labelme-json>
- [6] Roboflow, What is YOLOv7 Pytorch TXT?, <https://roboflow.com/formats/yolov7-pytorch-txt>
- [7] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”, arXiv:2207.02696, 6 Jul 2022
- [8] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, arXiv:1506.01497, 6 Jan 2016
- [9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector”, arXiv:1512.02325, 29 Dec 2016
- [10] Khaled R. Ahmed, “Smart Pothole Detection Using Deep Learning Based on Dilated Convolution”, Sensors 2021, 16 Dec, ber 2021
- [11] yolov5, <https://github.com/ultralytics/yolov5>
- [12] Yanzhao Zhu, Welqi Yan, “Traffic sign recognition based on deep learning”, Multimedia Tools and Applications (2022), 7 March 2022