高解像度画像からの物体検出高速化手法の検討

洞井 晋一[1] 小林 美貴[1] 沼田 直樹[1]

概要：ニューラルネットワークを用いた物体検出技術は日々進化しており, 実用的なサービスへの導入も進み始めている. また, デジタルカメラの微細化が進み, 市販されているカメラやスマートフォンでも 8K 相当の 4000 万画素での撮影を可能にしている. このような 8K 相当の高解像度画像を用いて物体検出を行うことで, 遠距離にある物体や小さな物体であっても検出が可能になってきている. 一方で従来の物体検出技術では 8K 相当の高解像度画像は想定されておらず, ある程度処理しやすい大きさにリサイズされてから物体検出を行うため, 8K 相当で無ければ映らないような物体の検出を行うことが難しい. 本論文では高解像度画像を分割して物体検出することで小さな物体の発見を可能にした. 実験の結果, 分割することで起きる誤検知や検出漏れを防ぎつつ, 従来よりも数十倍の物体の検出を可能にしている. また, 単純な分割では検出に 30〜400 秒程度時間がかかっていたが, 前処理を加えることで検出率は低下しつつも数秒程度まで高速化することを可能にした.

A Fast Object Detection Method from High Resolution Images

SHINICHI DOI[1] MIKI KOBAYASHI[1] NAOKI NUMATA[1]

1. はじめに

GPU（Graphic Processor Unit）はゲームや３次元グラフィックスなどの処理に専門的に用いられてきたが, その並列計算性能の高さから一般的な並列処理にも活用されるようになった. その結果, 従来より研究されてきたニューラルネットワークが実用化できるレベルで実装可能であると注目され, 様々な分野への活用が進んでいる. 特に画像処理の分野では GPU の並列計算性能を十分に活用したニューラルネットワークの手法によって著しく性能が向上した. 物体検出の分野でもその影響は大きく, ニューラルネットワークを用いた手法では高速かつ高精度に画像から物体の検出が可能となっている.

こうした物体検出技術を用いた社会課題の解決が様々な分野で期待されている. 自動車産業の分野ではより安全かつ高精度な自動運転技術への応用として, 人間や自動車, 自転車や二輪車, 標識などの検出を行うことが期待されている. 小売業界では Amazon Go[1] のように無人販売への応用だけでなく, 万引きの発見や在庫の管理などにも応用が期待されている. 防犯においても, 公共の場で刃物や危険物の検出を行うことで異常事態の把握に応用するなどが考えられている.

しかし, こうした最新技術の社会適用にはコストの壁が大きく立ちはだかる場合が多い. 例えば自動運転技術を活用した無人タクシー事業の場合, 無人タクシーを運用するランニングコストがタクシー運転手の賃金より高くなると導入はなかなか進まない. また, 小売業界においてもシステム全体のランニングコストが店員に支払う賃金より安くなければならず, GPU を使用することによる電気代, クラウドを利用する場合はクラウド利用料と通信量などを合計すると小規模な店舗では利用が難しいという実態がある. 特に物体検出の対象となるエリアが広くなればなるほど数多くのカメラを設置する必要があり, カメラそのものの価格や設置工事費などの初期費用も無視することができない. レンズの焦点距離を短くし, 広角での撮影を行えば広い範囲を一台のカメラで撮影することができるが, 同じ画素数のセンサーを使っている場合は, 検出対象が画像のなかで小さく映ることになるため, 物体検出が困難となる.

一方でデジタルカメラに用いられている CMOS センサーの微細化が進んだことで安価なカメラで撮影できる写真の解像度は日々向上している. スマートホンに搭載されてい

[1] 西日本電信電話株式会社

るカメラでも 2000 万画素を超えることは珍しくなく, 4000 万画素での撮影を可能にしている機種も存在している. 広角レンズを用いた撮影であってもこうした高画素センサーを有したカメラを利用することで, 検出対象となる物体も十分な画素数を保つことができるようになる.

しかし, 高画素の写真を用いて物体検出を行うことは容易ではない. まず, 画像そのものの情報量が多く, メモリへの負担が大きい. 4000 万画素の RGB 画像をそのままメモリ上へ展開した場合,1 枚あたり 114MB 程度消費することになる. そのため, そもそもニューラルネットワークを学習させることが難しく, 前処理によって 300〜1200 ピクセル程度にリサイズせざるを得ない.

本論文では,4000 万画素相当の画像に写った小さな物体を高速に検出する手法を提案する. 提案手法によって少ないカメラで広範囲の物体検出が可能になるため, 物体検出技術を実際に利用する場合に初期費用やランニングコストを抑えることが可能となる.

2. 関連研究

物体検出に CNN を用いた手法として R-CNN[2] がある. R-CNN では最初に物体候補となる領域を Selective Search を用いて抽出し, その領域に対して CNN を用いた物体検出を行う. しかし,R-CNN では Selective Search[3] による抽出に時間がかかるため, 高画素の画像を入力することは現実的ではない.

R-CNN を元に高速化した手法に Fast R-CNN[4] や Faster R-CNN[5] がある. Fast R-CNN では Selective Search を使用していることに変わりはなく, Faster R-CNN では Selective Search の代わりに RPN（Region Proposal Network）が用いられている. RPN では入力画像に対して何段かの畳み込みを行うが, 入力の画像サイズが大きくなるほど段数を増やす必要があり,4000 万画素相当の画像では GPU のメモリが足りずに学習ができなくなる.

YOLO[6] の手法では画像をいくつかのグリッドに分割して物体検出を行うがグリッド分割の前に正方形にリサイズを行う. 論文中では 448×448 にリサイズしており,Faster R-CNN と同様に 4000 万画素サイズを維持することは難しい.

文献 [7] ではこうした問題に対して,2 段階での検出手法を提案している. 1 段階目ではリサイズした画像に対して物体検出を行い, 2 段階目では 1 段階目で物体を検出した領域に対し, リサイズ前の画像を用いて再度物体検出を行う. この場合, 例えば道路上を人が集団で歩いている場合には, 1 段階目で誰か 1 人を発見するだけで集団全体を検出することができる. しかし,1 段階目では発見できないような小さい物体などは発見できない. また,2 段階目の物体検出に用いているグリッド分割の手法が, グリッドの境界上にある物体の検出やグリッドよりも大きな物体の検出に向いていない.

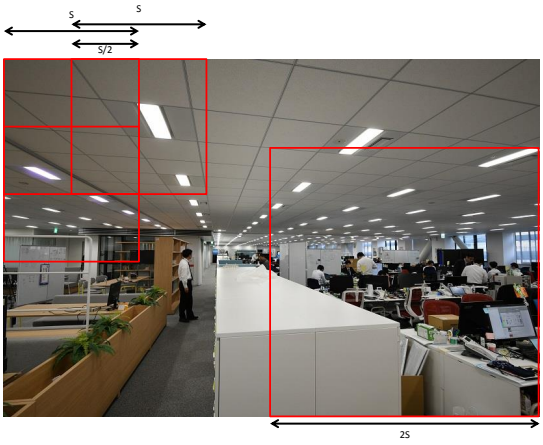

図 1　高解像度画像の分割手法

3. 提案手法

本論文では高解像度画像から物体検出を行う方法として下記の 3 つの手法を提案する.

3.1 小サイズに分割した画像の認識手法

高解像度の画像から物体検出を行う場合に, 既存の手法を活用するには入力画像を小さく分割することが確実である. 図 1 に分割の手法を示す. 赤枠で示したように一枚の画像を小領域で分割し, それぞれの画像に対して既存の物体検出手法を適用すれば, 高解像度画像にしか写らないような小さな物体まで検出できる. ここで, 元の画像サイズを w, h とし, 最小分割サイズを s とする. 分割領域を $(\mathrm{top}, \mathrm{left}, \mathrm{width}, \mathrm{height})$ で表すとする場合, 一番左上の分割領域は $(0, 0, s, s)$ で表すことが出来る. これを $A_{(0,0,s)}$ で表し,X 軸に x 個目,Y 軸に y 個目の領域を $A_{(x,y,s)}$ で表す. 分割画像の境界に対象となる物体が写っていた場合には検出ができない可能性があるため, 図 1 のように分割画像は $\frac{s}{2}$ ずつ重複させる. そのため, 例えば $A_{(1,1,s)}$ の分割領域は $(\frac{s}{2}, \frac{s}{2}, s, s)$ となる. また, 分割サイズよりも大きな物体を検出するために, 分割画像の縦横をそれぞれ 2 のべき乗倍にした画像についても検出を行う. 図 1 の右下には分割サイズを $2s$ とした場合の領域を示している. これらの画像は分割画像と同じ大きさにリサイズし, 物体検出を行う.

このように小サイズに分割した場合, 誤検出が増えるといった問題がある. 例えば図 2 に天井の照明を携帯電話と誤検出した例を示す. これは, 小サイズにした影響で物体の周辺にあるものの情報が失われてしまったため, 実際とは異なる物体を検出してしまったと考えられる. こうした誤検出を防ぐために, 小サイズで検出する物体は, その分割サイズの n%の割合に限ることとした. ここで n は 0 から 100 までの任意の数字を検出対象によって変更するパラメータ

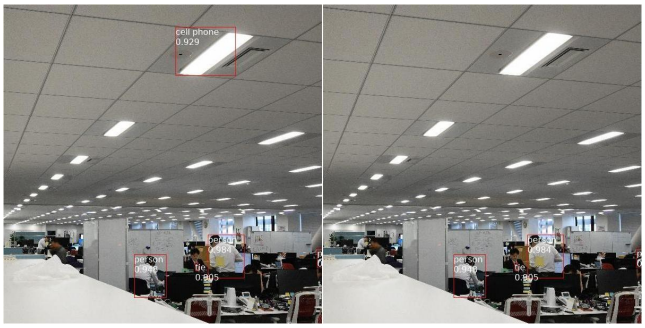

図 2　天井の照明を携帯電話と誤検出した例と対策後の検出例

図 3　物体の一部分だけで誤検出した例と対策後の検出例

図 4　閾値を 0.9 として物体検出を行った結果

とする．これは,n%よりも大きな物体であれば同じ領域を $2s$ の領域で物体検出した場合に同じように物体検出される可能性が高いためである．図 2 の右側に n = 30 とした場合の検出例を示す．携帯電話の誤検出を除外しつつ, 下に写っている人等の検出は問題なく行えていることが分かる．

また,2 のべき乗倍にした画像についても検出を行うため, 同じ画像を何度も検出することになり, 同じ物体を複数検出する場合がある．図 3 に同じ物体を複数検出した例を示す．左下の人物や右下のパラソルが複数検出されてしまっている．重なった同じ物体を排除する方法として IoU を計算する手法があるが, 分割画像の場合は片方が極端に小さい画像の場合があるため,IoU だけでは不十分である．図 3 の左側では IoU 値が 0.5 以上の場合は同一物体とみなしているが, 複数検出されてしまっている．そのため, 新しく発見した物体が既に発見した物体の内部にあるかどうかの判定によって同一物体かどうかを判別した．これにより, 例えば $2s$ で分割した画像では全体が人として検出され, s で分割した画像では人間の頭部だけが人として検出されるような場合であっても適切に除外できる．また, この手法を効率よく実施するために, 物体検出は分割サイズが大きい画像から行うこととしている．図 3 の右側に対策を施した検出結果を示す．人物とパラソルの両方で重複物体を排除できていることが分かる．

3.2　リサイズ画像の認識結果を用いてマスクを作成する手法

3.1 の方法では高い精度で物体検出を行えるが検出対象となる画像の枚数が増えるため物体検出に長い時間を要する．そこで, 文献 [7] の手法を参考にして物体検出の対象とする画像を限定する手法を提案する．高解像度の入力画像を 300〜1200 ピクセル程度の画像へ縮小し, 既存の物体検出手法を用いて物体検出を行う．図 4 に物体検出の結果を示す．文献 [7] ではこの検出結果を元にして実際に検出する分割エリアを特定するが, 図 4 の奥の方の人物は全く検出されていないため検出に失敗する可能性が高い．そこで,

小さな物体の検出精度を向上させるために物体検出に用いるスコアの閾値を低く設定する．図 5 に閾値を 0.3 まで下げた検出結果を示す．図に示すように閾値を下げることで奥の方の人物まで検出することが可能となるため認識の精度向上を期待できる．

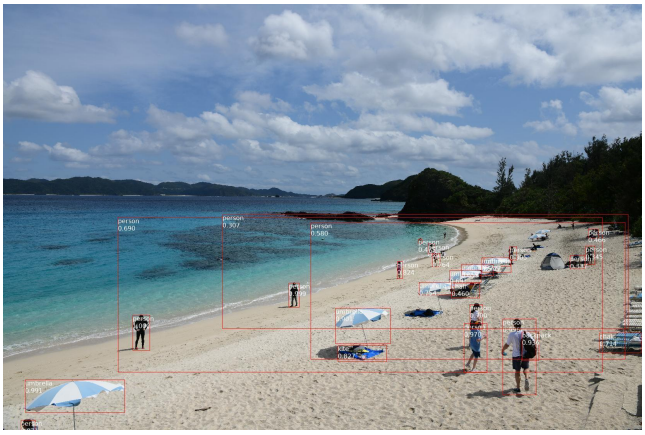

図 5　閾値を 0.3 として物体検出を行った結果

3.3　リサイズ画像から検出すべきエリアをニューラルネットワークに学習させる手法

3.2 の手法では 1 段階目の閾値を下げて検出を行った場合でも検出できない物体は発見できない．これはそもそも縮小したことで物体の情報が失われることで起こるため，縮小画像から物体検出を行う場合は避けることができない．一方で物体の周辺情報から物体が存在する可能性を知ることは出来る．例えば机や椅子がある場所や通路などには人がいる可能性が高く空や山には人がいる可能性が低い．こうした情報を 3.1 で検出した結果を用いて学習し，物体の存在する可能性が高い領域のみに物体検出を行う手法が考えられる．また，物体検出技術を異常検知に用いる場合，人がいる可能性が高い場所は物体検出せず，可能性が低い場所での物体検出を行うなどの応用が考えられる．本論文では 3.1 と 3.2 の検出結果からこのような物体検出の前段で行うニューラルネットワークの構築の有効性を確認する．

4.　評価

以下に提案手法の評価方法とその結果を示す．物体検出の手法として,Google 社の提供する Object Detection API[8] を利用し，Faster R-CNN,SSD[9],SSD-Lite,R-FCN[10] を用いてそれぞれの評価を行った．COCO データセット [11] によって学習したモデルを利用し，弊社オフィスを定点撮影した 138 枚の画像から人物の検出がどの程度行えるかを評価する．定点撮影は Nikon D850 を利用し，画像サイズは 8256 × 5504 で撮影した．分割する場合の最小サイズは 600 × 600 のサイズとし,8256 × 5504 の画像を 630 枚の画像に変換して物体検出を行った．また，検出に要した時間に関しても評価する．物体検出には nVidia 社の Titan X を 1 基使用し，バッチ処理等の並列化は行っていない．CPU として Core i9-7980XE，メモリは 128GB 搭載した計算機を利用している．

4.1　分割した画像を認識させる手法

3.1 に示した画像を分割する手法によって人物の検出精度がどの程度向上したかを評価する．結果を表 1 に示す．全てのモデルにおいて，分割して検出する手法は多くの物体を検出できており，Faster R-CNN では 20.4 倍,R-FCN では 54.1 倍,SSD では分割無しの場合にほとんど検出できていないが,1309.4 倍の物体を検出できたことになる．一方，検出時間は分割の有無に関わらず非常に長い時間がかかっており，分割した場合はしない場合に比べて検出時間

は 2 倍以上かかっている. 検出時間の内訳を調査したところ, これは画像の読み込みと Numpy 形式への変換に 20 秒以上かかっていることが分かった.

表 1 画像を分割して認識させる手法の評価

モデル名	平均物体検出数		平均検出時間（秒）	
	分割なし	分割あり	分割なし	分割あり
Faster R-CNN	4.485	91.427	1.364	400.251
SSD	0.007	9.166	0.795	30.214
SSD-Lite	0.014	8.775	0.793	31.821
R-FCN	0.869	47.057	0.858	54.958

4.2 リサイズ画像の認識結果を利用する手法

3.2 に示した手法によって, 人物の検出精度をどの程度保てたか, また時間がどの程度短縮できたかを評価する. 結果を表 2 に示す. マスクを施すことによって検出時間は大幅に短縮できているが, 検出結果は分割なしの場合と比較しても微増しているに過ぎない. 特に SSD の結果は分割なしの場合とほとんど変化が無いため, リサイズ画像の認識で閾値を下げても人物を検出できなかったことが分かる.

5. 考察

実験の結果から, 特に検出時間の短縮と, 検出精度の向上について考察する.

5.1 検出時間の短縮

分割の有無やリサイズ画像を用いてマスクする手法などを行っても 1 枚あたり 20 秒以上の非常に長い時間がかかることが分かった. これは主に 4000 万画素の画像の容量が非常に大きいことに起因していると考えられる. ただし,4000 万画素の画像を RGB の 3 チャネルで 8bit ずつ確保してもせいぜい 130MB 程度であり, 現在の CPU やメモリの性能を考えればそれほど長い時間がかかるとは考えにくく, 20 秒以上かかっているのは標準的なライブラリ等を組み合わせて使っていることが主要因と考えられる. Tensorflow の標準機能を使って読み込めるように実装を工夫するなどすれば十分に短縮できる可能性がある. 例えば R-FCN のモデルは 1 枚当たり平均して 92ms で検出できることが示されているため, 分割枚数が 630 枚であっても 58 秒程度で検出を終える可能性がある.

また,GPU のメモリに十分余裕があるのであれば, バッチ処理を実施することで大きく短縮できる可能性がある. バッチ数を 20 にすれば単純に 20 枚の画像を並列処理できるため,20 分の 1 程度まで削減を見込むことができる.

5.2 検出精度の向上

分割することによって検出精度を大きく向上させることが可能であると分かったが, 実際の社会導入を考えた場合, 出来る限り必要となるリソースを抑えて検出精度を向上させる必要がある. 高画素カメラを導入することでカメラの台数を減らし, 初期費用を低く抑えることが出来たとしても, 検出に GPU 等のリソースを多く使ってしまってはランニング費用が高くなり, 結果的にはカメラを多く設置する方がコスト安となってしまう恐れがある. 3.2 に示したように全体で検出を行った結果を用いてマスクすることで対象となる分割画像数を大きく減らすことはできたが, 検出精度はそれほど向上しなかった. これは, 閾値を下げて検出できる物体が少なかったことに起因しており, 論文 [7] の手法では限界があることを示している.

一方, 既存の手法では大きく精度向上できないということは, 3.3 で示したような手法が, 精度を大きく向上できる可能性を示している. 多くの画像データを学習させれば, 汎用的に物体が存在しそうな場所を学習できる可能性がある. 例えば机の上や人の近くにはペットボトルが存在しそうだが, 天井には存在しそうにない, といった形で学習できれば, 一般的な物体検出技術として高画質な画像を扱うことができるようになる. ただし, 実際の利用シーンではそこまでは求められておらず, 例えば監視カメラで人物を検出する場合であれば, 定点で設置されたカメラの画像を学習させ, そのカメラに人物が写り得る場所を学習できれば良い. 当然, 作業者がカメラ設置時にマスクする場所を設定することも考えられるが, 作業にかかるコストや人的ミス等を考えれば学習させる方法は大きな利点となりうる.

6. まとめ

本研究では 4000 万画素級の高画素画像を用いて物体検出を行う手法について示した. 画像を分割して物体検出する場合, 誤検知や検知漏れを防ぐために検出領域や検出後の処理を工夫する必要があることが分かった. また, 実際の写真を用いた実験では分割した物体検出手法が従来より数十倍の数の物体を検出できていることも分かった. 一方で分割することで検出には非常に長い時間がかかっており, 前処理によって検出対象となる分割領域を限定的にすることで時間の短縮を期待することができる. 今後は, 検出精度の向上を目指し,8K 相当の画像を利用した物体検出手法の確立を目指す.

参考文献

[1] Amazon go. https://www.amazon.com/b?node=16008589011.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[3] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer*

表 2　リサイズ画像での認識結果を用いてマスクする手法の評価（人物のみ）

モデル名	平均物体検出数			平均検出時間（秒）		
	分割なし	分割あり	マスクあり	分割なし	分割あり	マスクあり
Faster R-CNN	3.384	17.173	4.485	1.364	400.251	4.569
SSD	0.007	5.289	0.007	0.795	30.314	1.095
SSD-Lite	0.014	5.797	0.014	0.793	31.821	1.114
R-FCN	0.855	15.260	0.876	0.858	54.958	1.335

vision, Vol. 104, No. 2, pp. 154–171, 2013.

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[7] V ´ tR˚ uzicka, Franz Franchetti. Fast and accurate object detection in high resolution 4k and 8k video using gpus. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pp. 1–7. IEEE, 2018.

[8] Object detection api. `https://github.com/tensorflow/models/tree/master/research/object_detection`.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

[10] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pp. 379–387, 2016.

[11] Coco dataset. `http://cocodataset.org/`.