

Circle SSD を用いた FPGA 搭載ドローンでの農作物生育状況計測 Measurement of growing situation of agricultural crops on FPGA-mounted drone using Circle SSD

吉村 拓馬^{*1}
Takuma Yoshimura

^{*1} 株式会社ポコアポコネットワークス
poco-apoco Networks Co. Ltd.

In this paper, we report the optimization method for the case of introducing SSD which is one of the object detection models to the drones for the purpose of measuring the growth situation of agricultural crops as viewed from a bird's eye viewpoint. We have adopted a "Separable Convolution" and introduced "Circle SSD" which uses circular shape as a detection frame as an optimization method. As a result of the optimization, the F-measure 0.67 was realized with 14770 convolution params, and the object detection could be realized with limited calculation resources on the drone.

1. はじめに

本稿では、農作物の生育状況計測をドローンより上空から捉えた鳥瞰視点で行うことを目的とし、ドローンに搭載される FPGA(Field-Programmable Gate Array)ボードに物体検出モデルの 1 つである SSD(Single Shot MultiBox Detector)[Liu 16]を導入した事例について、物体検出モデルの小型化および円形の検出枠の適用に関して報告する。

ドローンとは遠隔操縦または自動操縦が可能な無人航空機の総称で、農業、設備点検、測量、警備、物資運搬など産業界での活用が広がっている[Nonami 18]。ドローンの多くは内蔵バッテリーによって駆動するため、長時間飛行させるためには消費電力を抑える必要がある。したがって生育状況計測のための搭載デバイスは軽重量で消費電力の小さいことが望ましく、さらに無線通信による電力損失を抑えることが求められる。

また本稿で採用する SSD は、統計的機械学習の数理モデルの 1 つで、画像認識や画像生成などに広く用いられる畳み込みニューラルネットワーク(以降 CNN)を用いている。しかし CNN は学習時、推論時に必要となる計算量が大きく、それゆえに高性能で消費電力の大きい CPU や GPU での運用が求められる。

本稿では、物体検出モデルの小型化を行い CPU、GPU に比べ消費電力の小さい FPGA[†]に SSD を導入した。ドローンから空撮した画像を搭載した FPGA で推論させることで、地上にあるサーバに文字列に比べデータ量の大きい画像を通信する必要がなく、通信量を抑えることが可能となる。全体の流れを図 1 に示す。本稿では図中のフロー 2 について述べる。

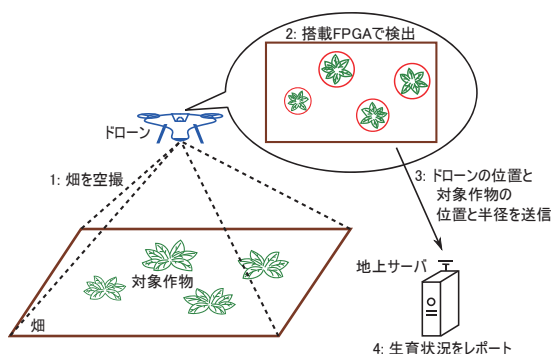


図 1 農作物生育状況計測の流れ

2. データセット

本稿では 2018 年の 5 月から 9 月にかけて撮影されたパイナップル耕作地の空撮画像 2187 枚および画像に含まれる、パイナップルの中心位置および葉の先端への半径のリストをデータセットとした。データセットの一部を以下の図 2 に示す。



図 2 パイナップル空撮画像 それぞれの株の中心部分から果実として流通しているパイナップルが生育する。

3. SSD の構成

本節では農作物の生育状況計測を実現するための、物体検出モデルである SSD の小型化、また円形の検出枠の適用について述べる。

3.1 SSD の構成要素

SSD は Extractor, Coder, Suppressor の 3 つの構成からなる(図 3)。Extractor は入力画像から畳み込み処理を行い物体認識に有効な特徴量を抽出、Coder は Extractor から得られた特徴量を各クラスの物体が存在する範囲の空間情報と確度(スコア)に変換、Suppressor は Coder から得られた検出枠候補から重複した確度の低い検出枠候補を除外する。

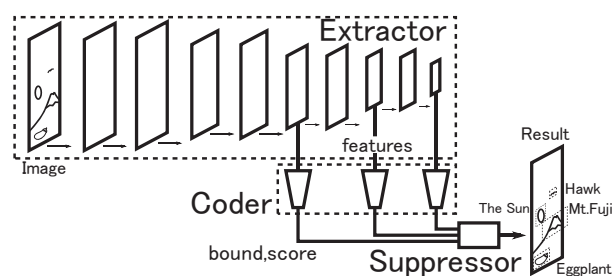


図 3 SSD の構成要素

連絡先: 吉村 拓馬, 株式会社ポコアポコネットワークス,
〒550-0012, 大阪府大阪市西区立売堀 2-1-9 日建ビル 6 階

[†] DIGILENT 社製 Zybo Zynq-7020, USB 給電 5V,
基板サイズ 88mm×122mm, BRAM 容量 630KB

3.2 FPGA のハードウェア制約

一般に高性能な CPU や GPU が要求される CNN を FPGA に実装するためには、記憶領域と論理ゲート数(回路規模)の制約条件を満たす必要がある。CNN の小型化の手法としてモバイルデバイス向け小型 CNN である MobileNet ベース SSD300(パラメータ数 6.8×10^6)[Howard 17]を参考とし、制約条件を満たすためパラメータ数を 1.5×10^4 までとし、Extractor の畳み込み層を 8 層まで減らす必要があった(表 1)。このため CNN の構成は一般のモデルに比べ制限される。例えば、このパラメータ数ではフィルタサイズ 3 チャンネル数 64 の、一般的な畳み込み層を 1 層も構成に入れることができない(パラメータ数 $3^2 \times 64^2 = 36864$)。

ただし、物体検出タスクとしての難度は先行研究に比べ低い。なぜなら先行研究のモデルは一般物体の複数クラスの認識精度を評価指標としている(COCO dataset[Lin 14])。本稿では、対象作物の 1 クラスのみであるうえに、対象の構造は水平に対し等方的、背景は主に赤褐色から黄褐色の土で色域が限られる。

表 1 モデルパラメータ

Model	Parameters	Dataset	Classes
deeplab-VGG	3.3×10^7	COCO	80
Inception V2	1.4×10^7	COCO	80
MobileNet	6.8×10^6	COCO	80
Ours	1.5×10^4	Pineapple	1

3.3 CNN の小型化

CNN は複数の処理層で構成され、一般に多くの演算を要する数値モデルである。1 つ目の処理層である入力層では画像に対して空間フィルタを適用(畳み込み処理)し、特徴マップを得る。以降の処理層では前の層で得られた特徴マップに対してさらに畳み込み処理し特徴抽出する、またはプーリング処理によりロバスト性を高める。こうしてより物体認識に有効な特徴量を抜き出した特徴マップを得る。この処理において物体検出モデルでは $10^9 - 10^{12}$ スケールの積和演算回数を要する[Howard 17]。

CNN の小型化・高速化において特徴抽出性能を保ちつつ、計算量を減らす構造がいくつか考えられている。代表的なものとして MobileNet[Howard 17]が導入した Depthwise Convolution, Pointwise Convolution, この 2 つを組み合わせた Separable Convolution がある(図 4)。その他にも SqueezeNet[Iandola 16]が導入した Fire Module, GoogLeNet[Szegedy 14]が導入した Inception Module があるが、実装上 FPGA の論理ゲート数を前記の構造より多く消費するため用いなかった。

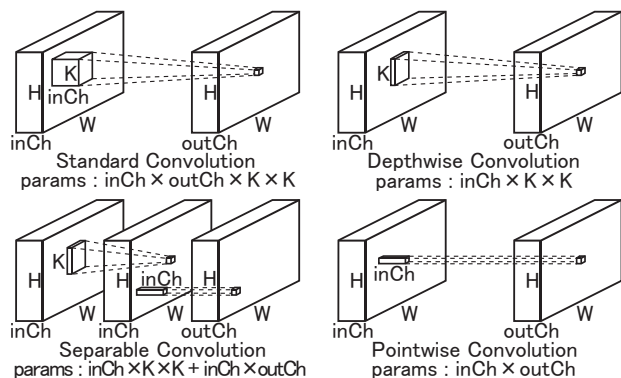


図 4 MobileNet における畳み込み処理 inCh:入力チャンネル数, outCh:出力チャンネル数, K:フィルタサイズ, W, H:特徴マップの幅および高さ, params: バイアスを除くパラメータ数(出力特徴マップ 1 座標あたりの積和演算回数に相当), なお図のストライド(S)は 1 である。

3.4 Extractor の構成方針

CNN は層が深くなるに従い、物体検出モデルにおいては一般にチャンネル数が増えていく。図 4 に示すとおり Standard Convolution, Pointwise Convolution, Separable Convolution のパラメータ数はチャンネル数が支配的であるため、チャンネル数が大きい処理層にはこれらの構造は避ける。本稿では、チャンネル数が最も小さい入力層には Standard Convolution, Coder に分岐するまでは Separable Convolution, 分岐後のチャンネル数が大きい処理層には Depthwise Convolution を用いた。

3.5 Coder の構成方針

Coder では Extractor で得られた特徴量を検出対象のクラスと背景クラスの確度、位置情報のチャンネルに変換するため、チャンネル間の畳み込み処理を要する。本稿では、最小限の構造である Pointwise Convolution を 1 層のみ用いた。また、一般的な物体検出モデルの検出枠は矩形であるが、植物のように水平等方向に空間を占める物体については円形であるほうが適していると考えた。矩形の検出枠では横位置、縦位置、横幅、縦幅の 4 パラメータの相対位置情報が必要となるが、円形の検出枠を適用することで、横位置、縦位置、半径の 3 パラメータのみで済み、さらに縦長横長の検出枠が不要となる。

3.6 Suppressor の構成方針

Suppressor では鳥瞰視点の空撮画像、円形を検出枠とする Circle SSD に適した検出枠重複除外処理に変更した。SSD[Liu 16]で用いられていた重複度ベースの除外方法を Circle SSD に適用すると、地表面面上に存在する検出対象の中心が重なることは無いにもかかわらず、円中心が近くに存在する半径の異なる複数の検出枠が得られてしまう。そこで本稿では互いの円の半径に対する中心への比距離が互いに 50%以下である場合、確度の低い方の円を重複した検出枠として除外することにした(図 5)。この閾値は検出対象の大きさと、近傍の検出対象との距離の比の分布から決定した。

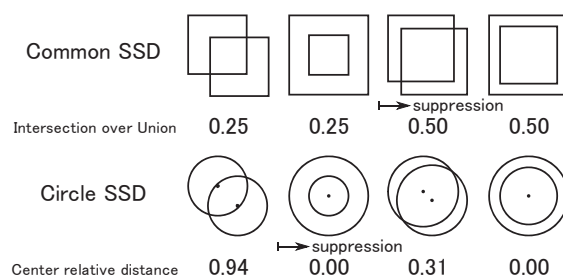


図 5 中心相対距離に基づく検出枠重複除外 上: 一般の SSD における検出枠配置とその Intersection over Union (IoU) 下: Circle SSD における検出枠配置と半径相対中心距離

3.7 Circle SSD の構成

本節をまとめると CNN の小型化をおこなった Circle SSD の構成は次頁の表 2 のようになる。列名は図 4 を参照のこと。

4. 学習時の変更点

本節では野外で運用するドローンに適したデータ拡張と、SSD[Liu 16]に円形の検出枠を適用したことによる特徴マップの形状の変更について述べる。

表 2 CNN の小型化をおこなった Circle SSD の構成

	Type	inCh	outCh	K	S	W	H	params
Extractor								11520
Conv1	StandardConv+Relu	3	8	5	1	512	288	608
Pool1	MaxPooling	8	8	2	2	512	288	0
Conv2	SeparableConv+Relu	8	16	5	1	256	144	352
Pool2	MaxPooling	16	16	2	2	256	144	0
Conv3	SeparableConv+Relu	16	32	5	1	128	72	960
Pool3	MaxPooling	32	32	2	2	128	72	0
Conv4	SeparableConv+Relu	32	64	5	1	64	36	2944
Conv5	DepthwiseConv+Relu	64	64	5	2	64	36	1664
Conv6	DepthwiseConv+Relu	64	64	5	2	32	18	1664
Conv7	DepthwiseConv+Relu	64	64	5	2	16	9	1664
Conv8	DepthwiseConv+Relu	64	64	5	2	8	5	1664
Coder								3250
Loc4	PointwiseConv	64	6	1	1	64	36	390
Conf4	PointwiseConv	64	4	1	1	64	36	260
Loc5	PointwiseConv	64	6	1	1	32	18	390
Conf5	PointwiseConv	64	4	1	1	32	18	260
Loc6	PointwiseConv	64	6	1	1	16	9	390
Conf6	PointwiseConv	64	4	1	1	16	9	260
Loc7	PointwiseConv	64	6	1	1	8	5	390
Conf7	PointwiseConv	64	4	1	1	8	5	260
Loc8	PointwiseConv	64	6	1	1	4	3	390
Conf8	PointwiseConv	64	4	1	1	4	3	260

4.1 データ拡張

野外での物体検出は屋内に比べ、天候や時刻による画像輝度の変動が大きいため難度が高い。ドローンの FPGA に導入する SSD の学習では画像の上下左右反転の他に、輝度の変動を考慮した以下のデータ拡張を行った。

- ガンマ値の変動(冪指数:0.5-1.25)
- RGB 輝度値の独立変動(倍率:0.82-1.00)

4.2 長方形画像の扱い

SSD の矩形の検出枠には縦幅横幅のパラメータがあり、画像の縦横縮尺比を変えることで、長方形画像であっても正方形画像に変形させ扱うことができる。しかし Circle SSD の円形の検出枠の尺度は半径のみであるため、画像変形後に縦長横長となる物体範囲を正しく検出できない。そこで画像の縦横縮尺比を変えず特徴マップの横幅と縦幅を変えることで、本稿で扱う空撮画像のような長方形画像(アスペクト比 16:9)に対応させた。

5. 推論時の最適化

本節では CNN を FPGA に導入するための計算量削減の手法について述べる。

5.1 固定小数点数演算

学習時に用いた浮動小数点演算器を並列動作させる GPU とは対照的に、推論時に用いた FPGA は浮動小数点演算器を持たないまたは乏しい。そのため浮動小数点数の演算は固定小数点数を用いたときに比べ、多くの論理ゲートを消費し回路規模の増大につながる。そこで本稿では符号 1bit 整数 5bit 小数 10bit の 16bit ビット幅の固定小数点数で推論を行った。なお、途中の積算は丸めず融合積和演算(FMA; fused multiply-add)とすることで誤差を小さくした。

5.2 バッチ正規化

畳み込み層と直後のバッチ正規化層[Ioffe 15]は以下の式のように、推論時に単一の等価な畳み込み層へ変換することができ、本稿ではこの変換式を用いた。ここで $\mathbf{x}$ は特徴マップ、 BN はバッチ正規化関数、 $Conv$ は畳み込み関数、 $\mathbf{W}$, b は畳み込み層の重みおよびバイアス、 γ , β はバッチ正規化が学習する尺度および位置パラメータ、 σ , μ は $Conv(\mathbf{x})$ のバッチ内標準偏差および平均の移動平均、 ϵ はゼロ除算を避けるための微小数である。出力チャンネル方向の各要素について記述している。

$$\begin{aligned}
 BN(Conv(\mathbf{x})) &= \gamma \frac{Conv(\mathbf{x}) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \\
 &= \gamma \frac{(\mathbf{W}\mathbf{x} + b) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \\
 &= \frac{\gamma\mathbf{W}}{\sqrt{\sigma^2 + \epsilon}}\mathbf{x} + \frac{\gamma(b - \mu)}{\sqrt{\sigma^2 + \epsilon}} + \beta.
 \end{aligned} \quad (1)$$

$$Conv'(\mathbf{x}) = BN(Conv(\mathbf{x})) = \mathbf{W}'\mathbf{x} + b',$$

$$\text{s.t. } \mathbf{W}' = \frac{\gamma\mathbf{W}}{\sqrt{\sigma^2 + \epsilon}}, b' = \frac{\gamma(b - \mu)}{\sqrt{\sigma^2 + \epsilon}} + \beta. \quad (2)$$

6. 結果

対象作物を本稿の Circle SSD で検出した結果画像を次頁の図 7 に示す。撮影高度、画像輝度に依らず対象作物を検出する事ができたものの、対象作物でない雑草への誤検出や、高高度の画像において対象作物の半径の誤認識が見られた。

次に Extractor の構成を表 1(パラメータ数 14770)、Mobile Net like(表 1 の Depthwise Convolution を Separable Convolution に置き換え、パラメータ数 31410)、VGG like(表 1 の Depthwise Convolution, Separable Convolution を Standard Convolution に置き換え、パラメータ数 481026)にした Circle SSD について、検出確度閾値を 0.10-0.95 に変化させたときの Precision-Recall 曲線、最大 F 値で評価を行った。検出枠と正解枠が一致したと判定する基準は互いの円の半径に対する中心への比距離が 25% 以内かつ半径比が 0.8-1.25 に収まっているかとする(図 6)。適合率(Precision)は一致枠数 / 検出枠数、再現率(Recall)は一致枠数 / 正解枠数、F 値は適合率と再現率の調和平均で計算される。また学習画像と検証画像は 9:1 に分け学習と検証を行った。

なお原論文のモデルは、本稿の対象作物 1 クラスのみの検出タスクに対し CNN のパラメータ数が多すぎ、過剰適合に陥ったため対照群としなかった。

Precision-Recall 曲線の比較、モデルサイズと最大 F 値の比較を次頁の図 8, 9 に示す。本稿のモデルの最大 F 値は 0.67 となり、32.5 倍のパラメータ数の VGG like, 2.1 倍のパラメータ数の MobileNet like に対し最大 F 値は 0.10, 0.02 の低下に収まった。また同じ空撮画像に対しモデルを変更した結果を図 10 に示す。

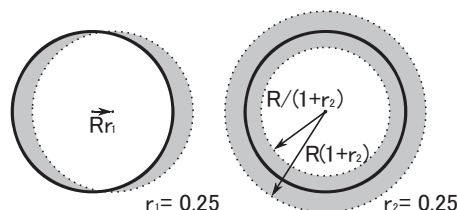


図 6 半径 R の正解枠に対して一致したとみなす検出枠の範囲



図 7 検出結果 撮影高度、画像輝度に依らず検出できている一方で、雑草への誤検出、高高度の画像での半径の誤認識が精度を落としている。なお、検出枠はランダムに着色している。

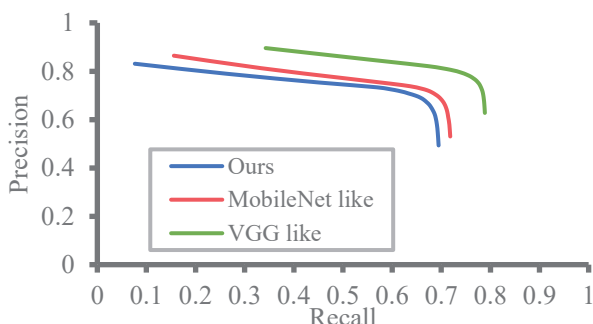


図 8 Precision-Recall 曲線の比較 理想的な検出器は Precision = 1, Recall = 1 の点を通る。F 値はいずれも検出確度閾値 0.60 で最大となった。

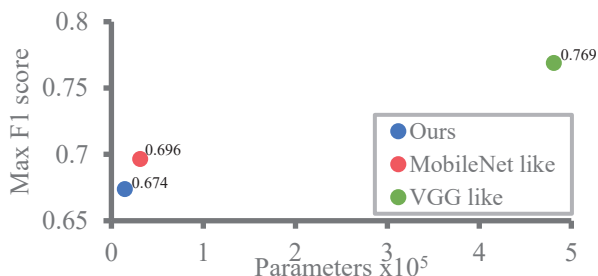


図 9 モデルサイズと最大 F 値の比較 モデルサイズと検出精度は一般的にトレードオフの関係にある。VGG like, MobileNet like のパラメータ数は本稿のモデルの 32.5 倍, 2.1 倍あり, 最大 F 値は本稿のモデルより 0.10, 0.02 大きい。



図 10 モデル変更時の検出結果の比較 この画像では VGG like と本稿のモデルは株が重なっている箇所にそれぞれ誤検出が 2 箇所, 1 箇所あり, VGG like は隣接する株に最長の葉が被っている株であっても検出範囲が正確である。

7. まとめ

本稿では鳥瞰視点で捉えた農作物の生育状況計測を目的に、円形の検出枠を適用した SSD として Circle SSD を提案した。またドローンに搭載する FPGA のハードウェア制約のもと、検出対象が 1 クラスのみという限られた条件付きであるものの、パラメータ数 14770 という小型 CNN 物体検出モデルを実用範囲内で実現することができた。

改良点として、Circle SSD への損失関数の最適化に、下式の円の類似度の採用を考えている。この類似度を用いることで株の重複箇所への誤検出が抑えられると期待される。ここで正解枠と Coder の各ポイントに割り振られたデフォルト枠の 2 円間について、 d は 2 円中心間距離、 r_1, r_2 は 2 円の半径である。

$$S = \max(0, 1 - \max(d/r_1, d/r_2)) \cdot \max(0, 1 - |\log(r_1/r_2)|) \quad (3)$$

課題として、矩形を検出枠とする SSD と本稿が提案した円形を検出枠とする Circle SSD では、検出枠が正解枠と一致したとみなす基準が異なり、精度の比較ができない事が挙げられる。両者で共通して用いることのできる評価手法について考察を深めたい。

最後に、ハイエンドな計算機で運用される高性能の CNN だけでなく、近年では個人向け PC やモバイルデバイス、エッジデバイスで運用される中性能・小型の CNN が求められている。物体検出タスクだけでなく、単一画像超解像や領域抽出、波形認識などのタスクを実行できる小型 CNN モデルの開発に取り組んでいきたい。

謝辞

本稿を作成するにあたり、ご助言を頂いた岡山県立大学の有本和民氏、大手前大学の奥村紀之氏に心より感謝致します。

参考文献

- [Liu 16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector, arXiv:1512.02325, 2016.
- [Nonami 18] 野波 健蔵, ドローン産業応用のすべて—開発の基礎から活用の実際まで—, ISBN-13: 978-4274506840, 2018.
- [Howard 17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv:1704.04861, 2017.
- [Lin 14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, Microsoft COCO: Common Objects in Context, arXiv:1405.0312, 2014.
- [Iandola 16] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, arXiv:1602.07360, 2016.
- [Szegedy 14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going deeper with convolutions, arXiv:1409.4842, 2014.
- [Ioffe 15] Sergey Ioffe, Christian Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv:1502.03167, 2015.