



HOKKAIDO UNIVERSITY

Title	ブルーベリー栽培高度化のためのマシンビジョンによるモニタリング自動化に関する研究
Author(s)	松岡, 和輝
Degree Grantor	北海道大学
Degree Name	博士(農学)
Dissertation Number	甲第15759号
Issue Date	2024-03-25
DOI	https://doi.org/10.14943/doctoral.k15759
Doc URL	https://hdl.handle.net/2115/94559
Type	doctoral thesis
File Information	MATSUOKA_Kazuki.pdf



ブルーベリー栽培高度化のためのマシンビジョンによる
モニタリング自動化に関する研究

北海道大学大学院農学院

農学専攻／生産フロンティアコース 博士後期課程

松岡 和輝

目次

第 1 章	諸言	5
第 2 章	物体検出 AI による花芽と果実の分類と計数	11
2.1	概要	11
2.2	材料と方法	11
2.2.1	画像データの取得および教師データの作成	11
2.2.2	教師データ作成用の Android アプリの開発	13
2.2.3	物体検出 AI	15
2.2.4	物体検出 AI の学習	16
2.2.5	物体検出 AI の評価方法	19
2.2.6	物体検出 AI の圃場管理への応用可能性	22
2.3	結果と考察	23
2.3.1	花芽検出 AI の結果	25
2.3.2	果実検出 AI の結果	33
2.3.3	花芽数を用いた収量予測への応用	41
2.3.4	結実率の可視化	43
2.3.5	成熟果実率の分布および成熟果実率の推移の可視化	44
2.4	結論	47

第 3 章	画像を用いた果実の推定についての検証	48
3.1	概要	48
3.2	材料と方法	48
3.2.1	画像の撮影及び真値の取得方法	48
3.2.2	葉の被覆率による果実数推定への影響の検証	49
3.2.3	撮影方角および両側撮影による果実数推定への影響の検証	53
3.3	結果と考察	54
3.3.1	画像に映る果実数と実際の個数との関係	54
3.3.2	葉の被覆率による果実数推定への影響の検証	56
3.3.3	撮影方角および両側撮影による果実数推定への影響の検証	61
3.4	結論	67
第 4 章	果樹トラッキングシステムの開発	68
4.1	概要	68
4.2	材料と方法	68
4.2.1	Robot Operation System (ROS)	68
4.2.2	データ取得システム	70
4.2.3	果樹検出 AI の作成	72
4.2.4	トラッキングアルゴリズム	73

4.2.5	果樹画像取得ソフトウェア	76
4.2.6	データの取得	77
4.2.7	評価方法.....	78
4.3	結果と考察	79
4.3.1	果樹検出 AI の評価.....	79
4.3.2	果樹トラッキングの評価.....	82
4.3.3	果実数と成熟果実率のマップ化.....	86
4.4	結論	87
第 5 章 総括		88
謝辞.....		91
参考文献.....		92

第 1 章 諸言

ブルーベリーは秋頃から花芽を形成し、翌年の春頃に花芽から複数の花を咲かせる低木果樹作物である。開花後は昆虫を介して受粉が進行し、受粉に成功した花のみ結実する。ブルーベリーは環境の変化に対する感受性が高いことから、栽培する際には、花芽の形成や、果実の結実および熟成について圃場の細かなモニタリングと管理が重要である。しかし、ブルーベリーは果樹 1 本あたり数百から数千もの花芽と果実をつけるため、人手による圃場全域のモニタリングには限界がある。そこで、花芽と果実を自動で認識・計数する技術の確立による、圃場モニタリングの自動化が求められている。

こうした圃場モニタリングの自動化には、物体検出と呼ばれる画像認識技術が有効だと考えられる。物体検出とは画像中に映る物体のクラス分類と位置を特定する技術である。近年、学習により自動でデータの特徴抽出が可能とする人工知能（AI）の一種である深層学習を用いた物体検出 AI が著しく進歩し、広く普及し始めている。農業分野においても物体検出 AI は応用されており、りんご果樹園における果実検出などに利用した応用例が報告されている（Tian et al. 2019）。

物体検出 AI による花芽の計数および分類が可能になることで、ブルーベリー圃場のモニタリング効率が大幅に上昇することが期待される。日本で広く栽培

されている高木性のハイブッシュブルーベリーにおいては、花芽数と果実数には相関があることが示されており (Salvo et al. 2012)、花芽数を把握することでその年の収量を予測し、必要な労働力を事前に算出できる可能性がある。さらに、冬季間の低温状態により発生した花芽の凍害を検出し、春先に剪定すべき果樹を事前に把握できる可能性がある。また、開花時期に蜂の巣箱を設置し、収量増加のために受粉促進を行うことが一般的である (Arrington and DeVetter 2018)。果樹ごとの花の総数に対して結実した果実数を求めることで蜂による受粉の効果を監視できると考えられるが、蜂は音に敏感であるため (Towne and Kirchner 1989)、機械を用いた作業を避ける必要がある。そこで、開花時期の自動モニタリングを避けるために花芽を花の代替として使用し、結実率 (果樹ごとの花芽の数に対する結実した果実数の比率) を求めることで、相対的に受粉の効果を監視できると考えられる。そのため、圃場全域にわたって結実率を自動推定し、結実率マップの作成が可能になれば、蜂の巣箱の配置が適切であるか調査できる可能性がある。一方で、物体検出 AI を使って花芽数の自動計数を実現するには、解像度に関する課題が考えられる。実際のブルーベリー圃場での実装を考慮すると果樹ごとに花芽を計数する必要があるが、一本の果樹が一枚の画像に収まるように撮影するためには、カメラを果樹から遠ざけなくてはならない。これにより、花芽の解像度が相対的に低下し、物体検出 AI の検出精度に悪影響を与え

る可能性が考えられる。しかし、一本の果樹が収まる画像からブルーベリーの花芽を計数および分類するための最適な解像度についてはまだ十分に検討されていない。

また、ブルーベリー果実の計数および分類の自動化も同様に圃場管理において有用であると考えられる。ブルーベリーの果実は同一の果樹内においても一斉には成熟しないため、同じ年に同一果樹で複数回にわたって収穫するのが一般的である。そのため、成熟果実が多い地点から順に収穫を行うことや、圃場全域の成熟果実数の推移を予測して収穫適期を判断する必要がある。現状、これらは人手による成熟果の監視から定性的に判断される。この判断を定量的に行うためには、物体検出 AI を利用して果樹ごとの成熟果や未成熟果の自動計数が有効であると考えられる。自動計数により圃場内の成熟果実率（果樹ごとの果実の総数に対する成熟果実数の比率）の分布や、圃場全域の成熟果実数の傾向を把握することで、データに基づいた収穫順序および収穫適期の適切な判断が実現されることが考えられる。ハイブッシュブルーベリー果実の検出および分類に物体検出 AI を用いた例として Mask R-CNN（He et al. 2018）というモデルを使用して、 1024×800 ピクセルと 1920×1080 ピクセルの画像からブルーベリー果実の検出および分類が可能であることが報告されている（Gonzalez et al. 2019）。しかし、この研究では近距離から撮影された画像を使用しており、花芽の場合と同様

に、一本の果樹が収まる画像からブルーベリーの果実を計数および分類するための最適な解像度については十分な検討がなされていない。

さらに、画像を用いた果実数の推定には、解像度に加えて葉による被覆と撮影方角を考慮する必要がある。結実後の果樹では葉が生い茂っており、多くの果実が葉に隠れてしまうために画像を用いた果実数の推定への悪影響が懸念される。葉に隠れた果実の推定に関する研究例として物体検出 AI により低木生のローブッシュブルーベリーの果実を計数し、収量を予測可能であることが報告されている (MacEachern et al. 2022)。しかし、樹形の異なるハイブッシュブルーベリーでの検討はされていないことに加え、葉の被覆率を考慮することで隠れた果実を含めた樹木全体の総果実数をより高精度に推定できる可能性がある。また、果樹の撮影方角の違いにより、同一果樹においても果実の見え方が異なる可能性がある。ブルーベリー果樹は列状に栽培されるのが一般的であるが、樹列の左側または右側にかかわらず、片側のみを撮影するだけで果実数を推定できるのが望ましい。しかし、樹列の左側・右側の日当たりの違いなどによって葉の繁茂度合いが異なり、撮影方角によって、果実数推定の結果にも影響する恐れがある。果樹に対する撮影方向の非対称性に関する研究では、物体検出 AI を用いてオレンジ果樹に結実している実際の果実数を推定する際、複数方向からの果樹画像を使用することで、その推定精度の向上することが報告されている (Mirhajia et

al. 2021)。しかし、ブルーベリー果樹の葉の被覆率、撮影方角、樹列の両側撮影の必要性に関する報告は存在せず、これらが果実の自動認識の精度に与える影響は十分に検討されていない。

物体検出 AI を用いて花芽や果実の自動計数を行うためには、果樹ごとの画像を自動取得することが求められる。ブルーベリー圃場では果樹を列状に植えるのが一般的であり、果樹同士が隣接して重なり合う場合が多く存在する。そのため、1 枚の画像には複数の果樹が映り込むことがあり、画像から 1 本の果樹を認識し切り抜く必要がある。また、人手で静止画の撮影を行うのは作業量の観点で困難であるため、圃場モニタリングの自動化は作業車などで樹列間を走行し、取り付けたカメラで撮影された動画を解析することで実現されと考えられる。果樹の認識には物体検出 AI の利用が有用であると考えられるが、動画に物体検出を適用するためには、検出した果樹を追跡しフレームごとに紐づける必要がある。動画に対して物体検出を適用した先行研究としては、根菜を検出しトラッキングにより個体を追跡して自動計数を行った研究が報告されている（板倉ら 2022）。この研究ではコンベア上で移動する根菜類のトラッキングを高精度に行っているが、果樹の検出に適した手法を検討する必要がある。

本研究では、ブルーベリー圃場におけるモニタリングの自動化を目的として、物体検出 AI を用いて画像上の果樹ごとの花芽と果実を認識し、それらの真値を

推定する手法を構築した。はじめに、正常芽および凍害芽の検出と分類を行い、花芽の検出に適切な画像解像度について検討した。次に、未熟果および成熟果の検出と分類を行い、花芽と同様の画像解像度に関する検討を行った。加えて、結実期の葉が果実の検出に影響を与える可能性を考慮し、果樹の葉による被覆、撮影方角、両側撮影の必要性についての検証を行った。そして、実際のブルーベリー圃場における応用例として、推定された花芽数からの実際の果実数の予測、推定された花芽数と果実数の組み合わせによる結実率算出、推定された成熟と未熟の果実数の組み合わせによる成熟果実率算出を行い、早期（花芽期）での収量予測、蜂による受粉効果の空間的変動の把握、果実の収穫適期・収穫順序の判断など農業現場への適用可能性を検討した。また、果樹ごとに物体検出 AI を適用し花芽や果実の自動計数を実現するために、果樹ごとの画像を取得するためのシステム開発を行った。

第2章 物体検出 AI による花芽と果実の分類と計 数

2.1 概要

この章では物体検出 AI の学習を行い、花芽検出モデルと果実検出モデルをそれぞれ作成し、果樹ごとの花芽数と果実数の分類および計数についての検討を行った。花芽については正常芽および凍害芽の検出と分類、果実については未熟果および成熟果の検出と分類を様々な解像度で検証した。その後、単回帰式を用いて画像中の花芽数と果実数を推定した。

2.2 材料と方法

2.2.1 画像データの取得および教師データの作成

物体検出 AI モデルの学習および検証のために、花芽および果実が実っているブルーベリー果樹の撮影をそれぞれ行った。本研究ではハイブッシュブルーベリー



図 1 撮影した果樹の例

一品種である「ブルークロップ」を対象とした。撮影は北海道余市町にあるブルーベリー農園であるアイケイファーム余市で行い、Canon 製デジタル一眼レフカメラ EOS 50D および、SIGMA 製レンズ 18-125mm F3.8-5.6 DC OS HSM を使用して 5202×3464 ピクセルの解像度の画像を得た。北海道余市町にあるブルーベリー農園であるアイケイファーム余市で行った。花芽の撮影は 2023 年 5 月上旬に 1 回、果実の撮影は 2023 年 6 月下旬から 8 月上旬にかけて 6 回行い、1 本の果樹が 1 枚の画像に全て収まるよう撮影を行った（図 1）。

データセットには果樹部分を切り抜いた画像を使用した。花芽や果実のアノテーションは PC 用のソフトウェアである labelImg (HumanSignal 2023) を使用して花芽や果実を目視で判断し、バウンディングボックスと呼ばれる矩形で囲った。バウンディングボックスは中心座標 (x, y)、画像サイズに対する横幅と高さの比率である (w, h) および物体のクラス番号 n を用いた (n, x, y, w, h) の YOLO 形式で定義した。花芽については、正常な球形の花芽を Alive、サイズの小さい花芽もしくは紫色を呈した花芽（凍害を受けた花芽）を Dead とし



図 2 クラス分類の例

てクラス分類を行った。また、果実については、完全に青色の成熟果実を Mature、緑色や赤色が含まれている未成熟果実を Immature としてクラス分類を行った（図 2）。

果樹は等間隔に並べて列状に植えられており、撮影した画像には背景の奥に別の列の果樹の画像が写り込んでいたが、手前の果樹に対してのみアノテーションを行った。花芽および果実の画像に対してそれぞれ 48 枚、37 枚ずつアノテーションを行い、各クラスのボックスの総数は、正常な花芽が 6604 個、凍害を受けた花芽が 487 個、未成熟の果実が 20908 個、成熟した果実が 4546 個となった。それぞれのデータセットは学習用、検証用、テスト用の 3 つに分割した。花芽のデータセットは学習用が 28 枚、検証用が 4 枚、テスト用を 15 枚とし、果実のデータセットは学習用が 19 枚、検証用が 4 枚、テスト用を 15 枚とした。果実のデータセットに関しては、果樹の成熟果実率が偏らないように分割を行った。

2.2.2 教師データ作成用の Android アプリの開発

花芽や果実は画像中に数百ほど存在し、PC 上でアノテーション作業を行うのは非常に困難である。そこで本研究ではバウンディングボックスを。タブレットを用いてアノテーションするための Android アプリケーションを Kotlin という言語を用いて開発した（図 3）。このアプリケーションについての機能を図中の

番号と対応させて以下に示す。(1)画像データを入れたフォルダを指定する(2)

フォルダ内の画像をリストに表示し、アノテーションを行う画像を選択する。

(3) ファイルリストは視認性のために、表示と非表示を切り替えることが可能

である。(4) 選択中のファイル名は図の左上に表示させている。(5) バウンデ

ィングボックスに割り当てるクラス番号を変更することが可能である。(6) タ

ッチ操作により、対象物体のアノテーションを行う。画像中のバウンディングボ

ックスは YOLO 形式に変換し、テキストデータに書き込みを行う。(7) はラベ

ルの表示と非表示を変更するボタンである。ラベルが描画された画像と、元画像

を分けて保存しているため表示と非表示を高速に切り替えることが可能である。

(8) このボタンを押した後にバウンディングボックスの内部をタッチするこ

とで、画像中のボックスを削除するとともに、テキストファイル中のデータも同

時に削除する。(9) によって、アノテーションモードのオンとオフを切り替え

ることができる。

また、本研究では果樹一本の画像に対して物体検出 AI を適用することから、

対象とする果樹を画像中から切り抜く必要がある。そこでアプリケーション上

に、切り抜いた画像の保存と切り抜いた画像に含まれるバウンディングボック

スのデータを取得する機能を搭載した。この機能によって、一本の果樹が映る画

像のアノテーションデータを取得することができる。また、切り抜いた画像中の

クラスごとのバウンディングボックス数を取得することが可能であるため、画像中の花芽数や果実数の計数に利用可能である。

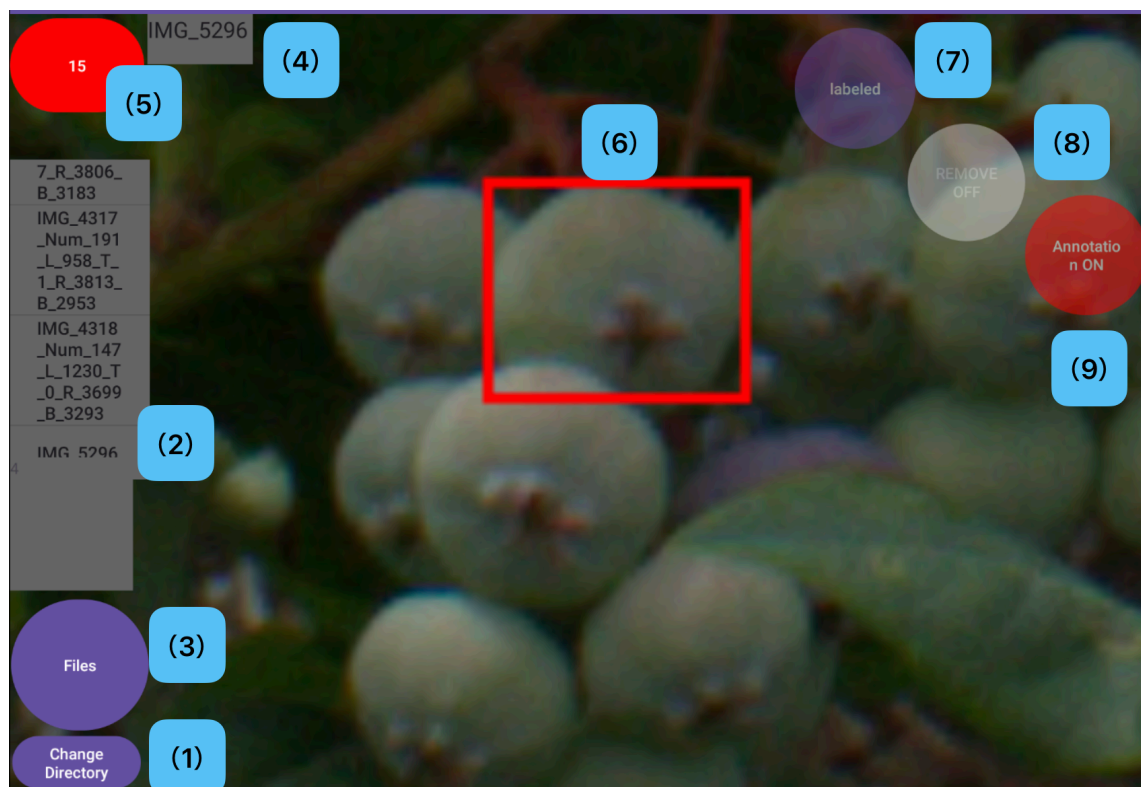


図 3 教師データ作成用の Android アプリの開発

2.2.3 物体検出 AI

物体検出 AI モデルとして YOLO (Redmon et al. 2016) の派生版モデルである YOLOv5 (Ultralytics 2023) および研究開始時点で最新モデルである YOLOv8 (Ultralytics 2023) を使用した。これらのモデルは画像データと YOLO 形式で表現されるバウンディングボックスが保存されたテキストデータにより学習可能であり、適切に学習を行うことで入力画像中に映る物体の検出とクラス分類

を行うことが可能である。また、AI の検出結果には信頼度を示す Confidence Score が同時に出力されており、この値が高いほど検出結果が正しい可能性が高くなる。Confidence Score の閾値は自由に設定することができ、AI による検出感度を調整することが可能である。YOLOv5 と YOLOv8 にはパラメータ数が異なる複数のモデルが存在しており、本研究ではそれぞれのモデルで最もパラメータ数が少ない YOLOv5n および YOLOv8n と、二番目にパラメータ数が少ない YOLOv5s および YOLOv8s というモデルを使用した。YOLO では一度入力画像を正方形画像に変換して学習および推論を行う。以下、表記を簡略化するために正方形に変換された入力画像の解像度の一辺のみを表記する。たとえば入力画像が 640×640 ピクセルの解像度であった場合は 640 ピクセルのように表記する。

2.2.4 物体検出 AI の学習

花芽用データセットおよび果実用データセットを使用して、花芽検出 AI および果実検出 AI の学習を行った。学習に使用した PC には GPU として GeForce RTX3090 (NVIDIA 社)、CPU として Intel Core i9-12900K (Intel 社) を搭載し、Ubuntu (20.04) 上において Python(3.8.10) および PyTorch(1.13.1+cu117) を用いてモデルの学習を行った。学習時の最適化アルゴリズムには SGD を使用した。また、エポック数を 2000 に設定し、50 エポックごとに早期学習停止の監

視を行った。バッチサイズについては GPU のメモリの制限により 8、4、2、1 のいずれか設定した。YOLOv5 および YOLOv8 は学習時の画像解像度を 32 の倍数に設定することが可能である。そこで本研究では YOLOv5n、YOLOv5s、YOLOv8n、YOLOv8s それぞれの公式事前学習済みモデルをベースとし、異なる解像度のブルーベリー画像（花芽、果実）で学習させた。学習後のモデルの重みパラメータについては、検証用データに対して最も評価が高いパラメータを採用した。以上の学習条件により、花芽検出 AI は合計で 12 個のモデル、果実検出 AI は合計で 10 個のモデルを作成した（表 1、表 2）。

表 1 花芽検出 AI の学習条件

モデル名	事前学習モデル	学習解像度	バッチ数
v5n640	YOLOv5n	640	8
v5n1280	YOLOv5n	1280	8
v5n2560	YOLOv5n	2560	8
v5s640	YOLOv5s	640	8
v5s1280	YOLOv5s	1280	8
v5s2560	YOLOv5s	2560	8
v8n640	YOLOv8n	640	8
v8n1280	YOLOv8n	1280	2
v8n2560	YOLOv8n	2560	1
v8s640	YOLOv8s	640	8
v8s1280	YOLOv8s	1280	2
v8s2560	YOLOv8s	2560	1

表 2 果実検出 AI の学習条件

モデル名	事前学習モデル	学習解像度	バッチ数
v5n640	YOLOv5n	640	8
v5n1280	YOLOv5n	1280	8
v5n2560	YOLOv5n	2560	8
v5s640	YOLOv5s	640	8
v5s1280	YOLOv5s	1280	8
v5s2560	YOLOv5s	2560	8
v8n640	YOLOv8n	640	4
v8n960	YOLOv8n	960	1
v8s640	YOLOv8s	640	4
v8s960	YOLOv8s	960	1

2.2.5 物体検出 AI の評価方法

学習によって得られたそれぞれのモデルを用いてテストデータに対して推論を行い、その結果について評価を行った。推論の際は一つのモデルに対して入力画像の解像度を 640、1280（果実検出 AI の v8s960 については 960）、2560 ピクセルに場合分けをして推論を行った。以下、推論を行う対象画像の解像度を推論

解像度として定義する。AI による検出結果の成功可否の判定基準として以下の式で定義される IoU (Intersection over Union) を用いた。

$$\text{IoU} = \frac{\text{検出結果と真値との交差領域}}{\text{検出結果と真値の和集合}}$$

本研究においては IoU が 0.5 を超えた場合のみ検出に成功したものとした。

AI による物体検出の評価指標 には平均適合率 (AP; Average Precision) および、F 値 (F1; F1-score)、適合率 (P; Precision)、再現率 (Re; Recall) に加えて決定係数 (R2; R-squared) を導入した。本研究では用いた画像中に映る花芽や果実の個数を推定することが目的であるため、テストデータに推論を行った際の AI による検出結果から画像に映る物体数の真値を推定する単回帰式を作成し、その際の決定係数 (R2; R-squared) を算出した。適合率、再現率、F 値、決定係数を評価する際の Confidence Score の閾値は、検証用データに対する推論結果の全クラスの F 値の平均値が最も高くなる値に設定した。適合率、再現率、F 値は以下の式で定義される。

$$P = \frac{TP}{TP + FP}$$

$$\text{Re} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot \text{適合率} \cdot \text{再現率}}{\text{適合率} + \text{再現率}}$$

ただし、TP は真陽性、NP は偽陽性、TN は真陰性、FN は偽陰性である。適合率は AI によって検出された物体のうち、真に対象とする物体の割合を示す指標

である。再現率は AI が検出する全ての物体のうち、どの程度検出に成功したかを示す指標である。F 値は再現率と適合率の調和平均で、この値が高いほど検出漏れや誤検知が少ないことを示す指標である。また、平均適合率の算出方法は以下の通りである。まず、検出結果を Confidence Score が高い順に並べて上位の検出結果から下位の検出結果に向かって正誤を求め、順に適合率と再現率をその都度計算する。その結果をグラフにプロットした PR 曲線と呼ばれる曲線と縦軸および横軸に囲まれる部分の面積が算出される平均適合率である。平均適合率は 0 から 1 の値を取り性能の高いモデルは再現率が大きい場合でも適合率が高い値をとることから、1 に近い値をとるほど精度の良い AI モデルであると評価できる。平均適合率に関しては花芽検出 AI および果実検出 AI に対してクラスごとに値を求め、評価を行った。検出 AI による推論は学習に使用した環境と同一のものを使用した。

また、実際のブルーベリー圃場に検出 AI を適用することを想定して前述の検出 AI を用いて画像に映る花芽数や果実数を推定し、その結果についての評価を行った。検出 AI による推論には、2560 ピクセルの高解像度画像および 640 ピクセルの低解像度画像を使用し、それぞれの推論解像度に対して前述の検出 AI の性能評価で明らかになった精度の高いモデルを選択した。まず、学習データに対して検出 AI を用いて画像中の物体数をクラスごとに検出し、その結果から画

像に映る物体数を推定するクラスごとの回帰モデルを単回帰分析により作成した。次にテストデータに対して、検出 AI を用いて画像中の物体数をクラスごとに検出し、作成した回帰モデルから画像に映る物体数をクラスごとに推定した。推定結果は RMSE（二乗平均平方根誤差）、MAE（平均絶対値誤差）、MAPE（平均絶対パーセント誤差）、MedAE（中央絶対誤差）の項目について評価を行った。

2.2.6 物体検出 AI の圃場管理への応用可能性

本研究で得られた物体検出 AI モデルの中で性能の高いものを使用して、花芽および果実の自動計数・分類のブルーベリー圃場管理への応用例を示した。

まず、発芽期における早期収量予測を想定して果樹ごとの花芽数と果実数の相関（Salvo et al. 2012）を、画像データを用いて同様の相関関係が得られるかについての検証を行った。同一果樹の発芽期と結実直後の画像を 17 本分用意し、果樹部分を切り抜いた画像から手作業と検出 AI を用いて花芽と果実の計数を行い、花芽数と果実数のピアソンの相関係数を算出した。

二つ目の応用例として、従来では大規模に取得が不可能であった花芽数と果実数の組み合わせによって得られる結実率（果樹ごとの花芽の数に対する結実した果実数の比率）のマップ化を行った。圃場内での位置を特定している 5 本の果樹の発芽期と結実直後の画像から、手作業と検出 AI による推定によって花芽数と果実数を計数し、結実率を算出した。Google マップ（Google 2023b）上

で果樹の位置座標を特定し、Google Cloud Platform の API (Google 2023a) を用いて Google マップ上に結実率のヒートマップとして表示させた。

最後に、適切な収穫順序や収穫時期の決定を目的として、成熟果実数と未成熟果実数の組み合わせから算出される成熟果実率（果樹ごとの果実の総数に対する成熟した果実の個数の比率）のマップ化と、成熟果実率および果実数の推移変化の可視化を行った。成熟果実率のマップ化については、結実率と同様の手法で成熟果実率のヒートマップを作成した。推移変化については 1 本の果樹を対象に画像から、果実の総数および成熟果実率を手作業と果実検出 AI による推定から取得し、それぞれを時系列順にプロットした。

2.3 結果と考察

YOLOv5n、YOLOv5s、YOLOv8n、YOLOv8s を 640、1280 (960)、2560 ピクセルの画像で花芽用および果実用に学習させたモデルを用いて、推論解像度 640、1280 (960)、2560 ピクセルの花芽画像および果実画像で推論を行った（図 4）。それぞれの推論において、適合率、再現率、F 値、平均適合率、決定係数の項目について評価を行った。以下、推論に使用したモデルと推論解像度を「モデル名・推論解像度」の形式で定義する。花芽検出と果実検出の評価についてはクラス間の平均適合率の平均値が大きい順に並べ替え、上位 10 個の結果をそれぞれ表 3 と表 5 に示した。



花芽



果実

図 4 AI による推論結果の例

2.3.1 花芽検出 AI の結果

花芽検出における Alive クラス（正常な花芽）の推論では平均適合率が 0.7 以上を示し、決定係数が 0.8 以上を示した（表 3 から表 6）。特に Alive クラスの推論結果の平均適合率は v8s2560・2560 が最も高く、次に v5s1280・1280、v5s1280・2560 の順に続いた。v5s1280 モデルは上位の結果に位置しており、v5s2560 モデルなどの高解像度画像で学習したモデルより優れた平均適合率を示した。v5s640・640 の低解像度画像を用いた結果においても平均適合率と決定係数がそれぞれ 0.544、0.802 となった。一方、Dead クラス（凍害を受けた花芽）の推論では、平均適合率の最大値は v8s2560・2560 であり適合率、再現率、F 値ともに約 30%と低い値を示した。Dead クラスの推論結果の決定係数は v8s2560・2560 が最も高く、次いで v5s2560・2560、v8s1280・2560 の順となった。これらの結果は推論解像度が 2560 ピクセルであり、Alive クラスの平均適合率や決定係数と比較すると低い結果となった。

表 3 花芽検出 AI の評価 (Alive クラス・YOLOv5)

モデル名	推論 解像度	Alive クラス				
		P	Re	F1	AP	R ²
v5n640	640	0.702	0.465	0.559	0.5	0.372
v5n640	1280	0.431	0.619	0.509	0.535	0.856
v5n640	2560	0.2	0.195	0.197	0.105	0.21
v5n1280	640	0.622	0.375	0.468	0.385	0.496
v5n1280	1280	0.711	0.69	0.701	0.699	0.903
v5n1280	2560	0.524	0.736	0.612	0.59	0.851
v5n2560	640	0.212	0.118	0.151	0.09	0.04
v5n2560	1280	0.652	0.651	0.652	0.637	0.91
v5n2560	2560	0.691	0.757	0.723	0.575	0.915
v5s640	640	0.671	0.529	0.591	0.544	0.802
v5s640	1280	0.495	0.676	0.572	0.611	0.824
v5s640	2560	0.21	0.2	0.205	0.094	0.428
v5s1280	640	0.656	0.44	0.527	0.466	0.633
v5s1280	1280	0.711	0.719	0.715	0.732	0.886
v5s1280	2560	0.557	0.761	0.643	0.707	0.828
v5s2560	640	0.404	0.185	0.253	0.174	0.183
v5s2560	1280	0.783	0.64	0.704	0.698	0.902
v5s2560	2560	0.767	0.729	0.748	0.658	0.909

表 4 花芽検出 AI の評価 (Alive クラス・YOLOv8)

モデル名	推論 解像度	Alive クラス				
		P	Re	F1	AP	R ²
v8n640	640	0.603	0.464	0.524	0.377	0.763
v8n640	1280	0.617	0.603	0.61	0.53	0.665
v8n640	2560	0.109	0.154	0.128	0.037	0.3
v8n1280	640	0.536	0.366	0.435	0.271	0.64
v8n1280	1280	0.73	0.63	0.676	0.554	0.931
v8n1280	2560	0.564	0.59	0.577	0.499	0.605
v8n2560	640	0.02	0.048	0.028	0.018	0.012
v8n2560	1280	0.692	0.511	0.588	0.495	0.404
v8n2560	2560	0.793	0.67	0.726	0.659	0.805
v8s640	640	0.552	0.416	0.474	0.295	0.531
v8s640	1280	0.544	0.539	0.541	0.424	0.427
v8s640	2560	0.043	0.154	0.067	0.015	0.574
v8s1280	640	0.625	0.453	0.525	0.417	0.658
v8s1280	1280	0.706	0.733	0.719	0.697	0.893
v8s1280	2560	0.59	0.742	0.657	0.633	0.722
v8s2560	640	0.021	0.059	0.031	0.136	0.107
v8s2560	1280	0.794	0.41	0.541	0.579	0.353
v8s2560	2560	0.77	0.746	0.758	0.755	0.924

表 5 花芽検出 AI の評価 (Dead クラス・YOLOv5)

モデル名	推論 解像度	Dead クラス				
		P	Re	F1	AP	R ²
v5n640	640	0	0	0	0.004	0.23
v5n640	1280	0.18	0.072	0.103	0.037	0.482
v5n640	2560	0	0	0	0.001	0
v5n1280	640	0	0	0	0.002	0.259
v5n1280	1280	0.241	0.109	0.15	0.059	0.32
v5n1280	2560	0.154	0.151	0.153	0.023	0.413
v5n2560	640	0.003	0.008	0.004	0	0.039
v5n2560	1280	0.121	0.101	0.11	0.04	0.119
v5n2560	2560	0.325	0.235	0.273	0.07	0.363
v5s640	640	0.286	0.034	0.06	0.033	0.313
v5s640	1280	0.189	0.059	0.09	0.031	0.303
v5s640	2560	1	0	0	0.007	0.001
v5s1280	640	0.081	0.025	0.038	0.004	0.144
v5s1280	1280	0.189	0.143	0.163	0.071	0.434
v5s1280	2560	0.188	0.143	0.163	0.066	0.5
v5s2560	640	0.005	0.008	0.007	0.001	0.148
v5s2560	1280	0.185	0.101	0.13	0.037	0.358
v5s2560	2560	0.327	0.294	0.31	0.099	0.566

表 6 花芽検出 AI の評価 (Dead クラス・YOLOv8)

モデル名	推論 解像度	Dead クラス				
		P	Re	F1	AP	R ²
v8n640	640	0	0	0	0.019	0
v8n640	1280	0	0	0	0.061	0
v8n640	2560	0	0	0	0.007	0
v8n1280	640	0	0	0	0.005	0
v8n1280	1280	0	0	0	0.061	0
v8n1280	2560	0	0	0	0.008	0
v8n2560	640	0	0	0	0	0.094
v8n2560	1280	0.2	0.025	0.045	0.011	0.093
v8n2560	2560	0	0	0	0.128	0
v8s640	640	0	0	0	0.006	0
v8s640	1280	0	0	0	0.034	0
v8s640	2560	0	0	0	0	0
v8s1280	640	0.182	0.017	0.031	0.011	0.202
v8s1280	1280	0.223	0.101	0.139	0.166	0.374
v8s1280	2560	0.427	0.151	0.223	0.173	0.509
v8s2560	640	0	0	0	0	0.019
v8s2560	1280	0.166	0.059	0.087	0.009	0.031
v8s2560	2560	0.329	0.277	0.301	0.386	0.598

精度評価の結果から、花芽検出 AI を用いて一本の果樹が収まった画像から正常な花芽の検出を高精度に実現した。さらに、作物は異なるがキウイの花芽の接写画像に対する先行研究 (Li et al. 2022) と比較して、遥かに花芽の解像度が低い 640 ピクセルの画像を用いて学習・推論を行った本研究の場合においても正常な花芽を検出可能であることが示された。また、v5s1280・1280 での推論結果から 2560 ピクセルの画像を用いた推論結果と平均適合率などの結果に大きな差がないことが明らかになった。物体検出 AI の学習と推論時の計算コストの削減には低解像度画像の使用が有効であり、正常な花芽の検出に関しては 1280 ピクセルの画像まで解像度を下げても良いことを示唆している。これにより、安価な GPU でも実用には十分であるといえるため、導入コストの低い物体検出 AI の実装が期待される。また、計算コストを抑えられることから、圃場内でのリアルタイム処理によるデータ取得が可能となることが期待される。

一方で、凍害の花芽の検出精度は低い結果となった。Dead クラスの推論における平均適合率が Alive クラスのそれに比べて非常に低かった要因は、推論の際に用いた Confidence Score の閾値を全クラスの平均 F 値が高くなる値に設定していたことが挙げられる。Confidence Score の閾値を下げることで Dead クラスの検出性能が向上する可能性がある。また、別の要因として Dead クラスの学習が Alive クラスほど進んでいなかったことも挙げられる。推論結果における決定

係数の上位が全て 2560 ピクセルであったことから、Dead クラスに対する検出結果を向上させるには、2560 の高解像度画像を使用した学習と推論が有効であることが示唆されている。高解像度画像が上位の結果となった要因としては、凍害を受けた花芽は紫色を呈していることで枝との色情報による差異が少ない点や、凍害を受けた花芽は成長しておらずサイズが非常に小さいためであると推察される。凍害の検出を行う場合は、ある程度の解像度が必要であることが示され、物体検出 AI の導入に関するコストが大きくなることを示唆している。

表 7 花芽検出 AI による予測値と真値との誤差

推論解像度・クラス	RMSE	MAE	MAPE	MedAE
高解像度・Alive	17.14	14.47	11.09	14.00
低解像度・Alive	18.27	13.18	10.79	11.00
高解像度・Dead	4.09	3.47	81.19	3.00
低解像度・Dead	4.83	4.00	49.18	4.00

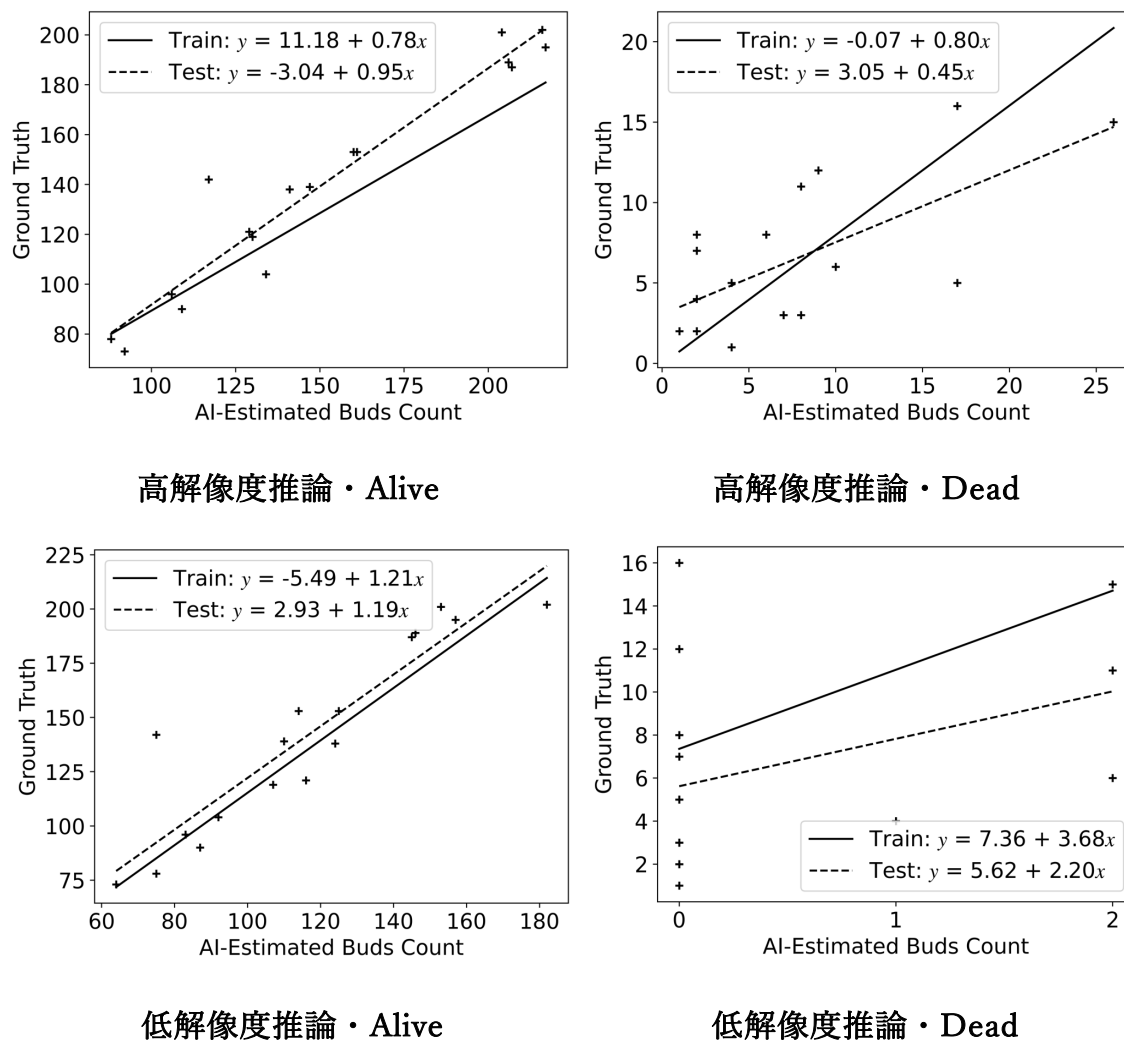


図 5 花芽検出 AI による画像に映る花芽数の推定

次に実際のブルーベリー圃場で花芽検出を行うことを想定して前述の v5s640・640 および v8s2560・2560 の花芽検出モデルを使用して、画像に映る花芽数の推定を行い、その精度を検証した（表 7）。花芽検出 AI を用いた個数推定の結果、Alive クラスにおいては解像度に関わらず MAPE が約 10%を示した。

ただし、高解像度推論における学習時のデータから作成した回帰式と、テストデータから作成した回帰式では乖離が生じた（図 5）ため、低解像度推論と誤差がほとんど変わらない結果となった。一方、Dead クラスでは解像度に関わらず大きな誤差が生じ、特に低解像度画像を用いた推論では Dead クラスの花芽数の推定は困難であった。この結果から花芽検出 AI を用いて、画像に映る正常な花芽数を高精度に推定可能であることが示された。また、高解像度推論において学習データで作成した回帰式の傾きがテストデータで作成した回帰式よりも小さかったことから、学習データに対する過学習の発生が考えられる。本研究では検証用データの不足により学習用データを使用して回帰式を作成したが、個数推定の精度を向上させるために今後より多くのデータセットを作成し、検証用データを用いて回帰式を作成することが推奨される。

2.3.2 果実検出 AI の結果

果実検出 AI による推論では Immature クラス（未成熟果）と Mature クラス（成熟果）のどちらの場合においても多くのケースで平均適合率が 0.7 や 0.8 以上を示し、決定係数も 0.8 以上を示した（表 8 から表 11）。特に v8s640・640 モデルでは、低解像度画像による学習と推論であるにもかかわらず、高い平均適合率や決定係数を示しており Mature クラスではそれぞれ 0.724、0.908 の値を示した。

表 8 果実検出 AI の評価 (Immature クラス・YOLOv5)

モデル名	推論 解像度	Immature クラス				
		P	Re	F1	AP	R ²
v5n640	640	0.005	0.009	0.007	0.003	0.494
v5n640	1280	0.064	0.191	0.096	0.051	0.513
v5n640	2560	0.004	0.008	0.005	0.002	0.253
v5n1280	640	0.325	0.119	0.174	0.128	0.232
v5n1280	1280	0.79	0.556	0.653	0.654	0.89
v5n1280	2560	0.695	0.63	0.661	0.634	0.933
v5n2560	640	0.14	0.065	0.089	0.055	0.102
v5n2560	1280	0.612	0.555	0.582	0.588	0.826
v5n2560	2560	0.853	0.712	0.776	0.807	0.973
v5s640	640	0.415	0.16	0.23	0.179	0.483
v5s640	1280	0.656	0.468	0.546	0.506	0.775
v5s640	2560	0.034	0.055	0.042	0.013	0.605
v5s1280	640	0.36	0.181	0.241	0.206	0.477
v5s1280	1280	0.836	0.609	0.705	0.712	0.896
v5s1280	2560	0.703	0.671	0.687	0.699	0.935
v5s2560	640	0.186	0.074	0.106	0.077	0.001
v5s2560	1280	0.675	0.58	0.624	0.595	0.772
v5s2560	2560	0.792	0.729	0.759	0.81	0.941

表 9 果実検出 AI の評価 (Immature クラス・YOLOv8)

モデル名	推論 解像度	Immature クラス				
		P	Re	F1	AP	R ²
v8n640	640	0.672	0.357	0.466	0.362	0.709
v8n640	1280	0.652	0.513	0.574	0.486	0.752
v8n640	2560	0.03	0.055	0.039	0.006	0.577
v8n960	640	0.71	0.335	0.455	0.397	0.437
v8n960	960	0.806	0.556	0.658	0.625	0.897
v8n960	2560	0.142	0.103	0.119	0.031	0.17
v8s640	640	0.749	0.414	0.533	0.451	0.768
v8s640	1280	0.662	0.551	0.601	0.498	0.804
v8s640	2560	0.036	0.048	0.041	0.008	0.594
v8s960	640	0.764	0.376	0.504	0.438	0.537
v8s960	960	0.886	0.55	0.679	0.691	0.932
v8s960	2560	0.194	0.168	0.18	0.094	0.469

表 10 果実検出 AI の評価 (Mature クラス・YOLOv5)

モデル名	推論 解像度	Mature クラス				
		P	Re	F1	AP	R ²
v5n640	640	0.026	0.052	0.035	0.015	0.289
v5n640	1280	0.045	0.065	0.053	0.025	0.33
v5n640	2560	0	0	0	0	0.002
v5n1280	640	0.551	0.334	0.416	0.328	0.757
v5n1280	1280	0.752	0.646	0.695	0.676	0.95
v5n1280	2560	0.652	0.649	0.651	0.578	0.805
v5n2560	640	0.34	0.184	0.239	0.165	0.212
v5n2560	1280	0.617	0.567	0.591	0.536	0.936
v5n2560	2560	0.88	0.767	0.819	0.815	0.927
v5s640	640	0.583	0.378	0.459	0.384	0.825
v5s640	1280	0.512	0.493	0.502	0.418	0.975
v5s640	2560	0.02	0.026	0.023	0.009	0.412
v5s1280	640	0.556	0.437	0.489	0.416	0.896
v5s1280	1280	0.774	0.664	0.715	0.691	0.965
v5s1280	2560	0.673	0.66	0.666	0.637	0.753
v5s2560	640	0.306	0.207	0.247	0.157	0.207
v5s2560	1280	0.699	0.372	0.486	0.468	0.644
v5s2560	2560	0.867	0.735	0.796	0.802	0.972

表 11 果実検出 AI の評価 (Mature クラス・YOLOv8)

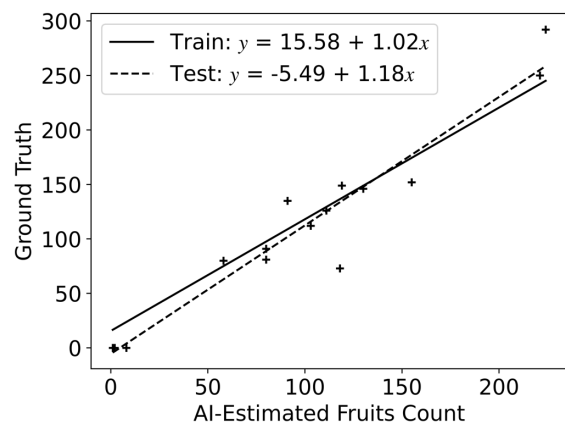
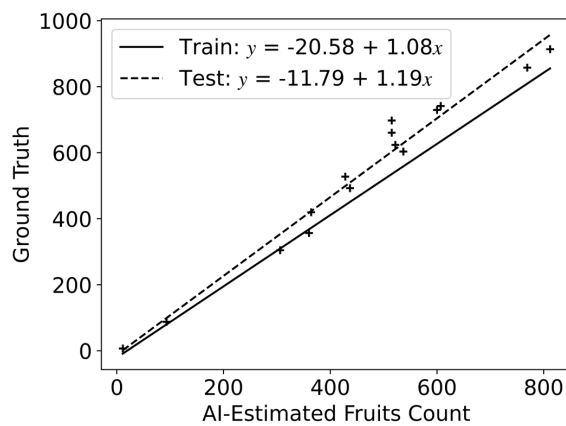
モデル名	推論 解像度	Mature クラス				
		P	Re	F1	AP	R ²
v8n640	640	0.729	0.554	0.63	0.652	0.901
v8n640	1280	0.502	0.512	0.507	0.433	0.759
v8n640	2560	0.023	0.037	0.029	0.005	0.341
v8n960	640	0.757	0.54	0.63	0.67	0.904
v8n960	960	0.772	0.653	0.708	0.783	0.93
v8n960	2560	0.121	0.173	0.142	0.027	0.779
v8s640	640	0.758	0.576	0.655	0.724	0.908
v8s640	1280	0.524	0.595	0.557	0.438	0.691
v8s640	2560	0.031	0.07	0.043	0.009	0.528
v8s960	640	0.713	0.605	0.655	0.715	0.869
v8s960	960	0.741	0.711	0.726	0.782	0.947
v8s960	2560	0.141	0.339	0.199	0.077	0.524

精度評価の結果から果実検出 AI を用いて、一本の果樹が写った画像から成熟果と未熟果の検出・分類を高精度に実現した。ブルーベリー果実の検出・分類に関する先行研究では、果実の房に接写して撮影し平均適合率の最大値は、1024×800 ピクセルの画像に対する推論で0.745であった(Gonzalez et al. 2019)。本研究では一本の果樹が収まる画像を使用したにも関わらず、果実の接写画像に

対する推論結果と同程度または超える結果が得られた。果実の解像度が低い本研究において、近接画像を使用した先行研究を超える結果が得られた要因としては、先行研究に使用したモデルよりも世代の進んだ性能の高いモデルを使用したためだと推察される。また、果実検出 AI では推論解像度が 640 の画像を使用した場合においても決定係数が 0.9 を超える推論結果が得られた。これにより、限られたコンピュータ環境下で学習されたモデルを用いた推論や、Raspberry Pi のような低計算リソースのコンピュータを用いた果実数の推定が実現可能であることが示唆された。また、低解像度画像が利用可能であることから花芽の場合と同様に、導入コストの低い物体検出 AI による圃場モニタリングの実装やリアルタイム性のあるデータ取得が期待される。

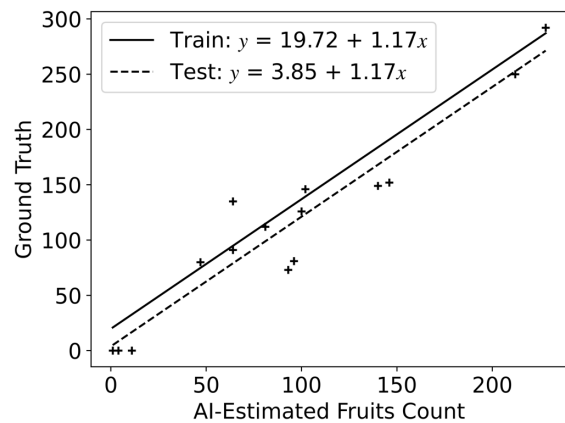
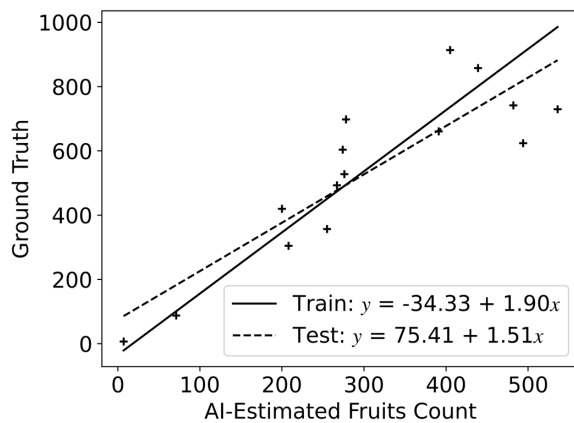
表 12 果実検出 AI による予測値と真値との誤差

推論解像度・クラス	RMSE	MAE	MAPE	MedAE
高解像度・Immature	63.33	77.96	23.84	49.00
低解像度・Immature	106.93	135.97	26.12	74.00
高解像度・Mature	18.73	24.76	30.47	17.00
低解像度・Mature	24.46	29.16	33.24	21.00



高解像度推論・Immature

高解像度推論・Mature



低解像度推論・Immature

低解像度推論・Mature

図 6 果実検出 AI による画像に映る果実数の推定

また、花芽の場合と同様に v8s640・640 および v5s2560・2560 の果実検出モデルを使用して画像に映る果実数を推定し、その精度を検証した（表 12）。個数推定の結果、花芽検出 AI と同様に学習時のデータから作成した回帰式と、テストデータから作成した回帰式が大きく乖離する場合は生じた（図 6）ものの、MAPE は 23.84-33.23%の範囲に収束した。

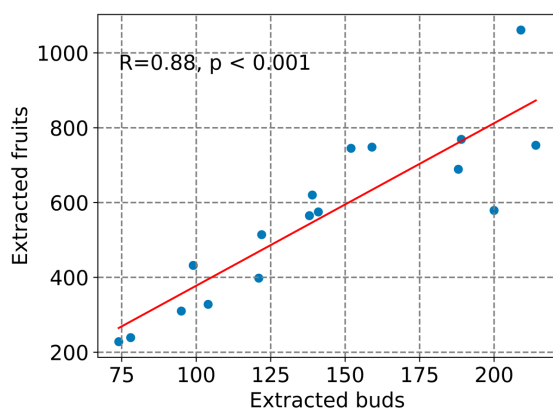
この結果から花芽検出 AI の Alive クラスの結果と比較して誤差は大きいもの

の、果実検出 AI を用いて、画像中の未成熟果実数および成熟果実数を推定することが可能であることが示された。また、Immature と Mature の両クラスにおいて高解像度推論による個数推定の結果は数%の MAPE 差で、低解像度推論による個数推定の結果を上回ることが示された。一方で花芽検出 AI の Alive の結果と比較すると MAPE で 10%以上の差が見られた。この要因として学習データに対する過学習に加え、テストデータが天候などの影響により学習データの画像と彩度やコントラストなどが異なるデータであったためだと考えられる。これにより、特に果実検出 AI の場合は学習データとテストデータの果実数の検出数に大きな差が生じてしまい、学習データから作成した回帰式とテストデータから作成した回帰式が大きく乖離したことが要因であると考えられる。一方で、物体検出 AI を用いてオレンジ果実の検出を mAP で最大 0.914 を達成し、全画像に映る総果実数を最大 91.45%の精度で計数した先行研究(Mirhajia et al. 2021)では、1568 枚のデータセットの 80%を学習データとしていた。先行研究ではデータ数が十分に確保されたために学習データに対する過学習を抑制し、果実検出 AI が様々なテストデータの画像に対応可能であったと推察される。本研究では果実数が 2 万を超えるデータセットであったが、学習に使用した画像は数十枚に留まり画像の枚数という観点でデータ数が不足していた。今後、画像数を十分増やしたデータセットによって学習された果実検出 AI の使用が推奨される。

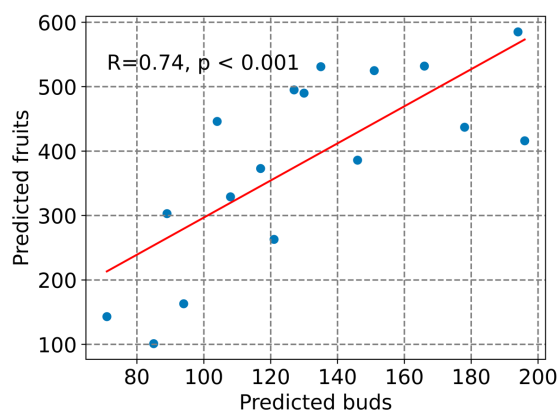
一方で、果実用のデータセット作成は果実数が非常に多いため、果実のアノテーションには果樹 1 本あたり 1 時間程度の時間がかかってしまう。これにより、十分な画像数の確保が困難である場合は、撮影環境がテストデータに近い検証用データを用いた回帰式の作成によって、より精度の高い回帰結果が得られることが期待される。

2.3.3 花芽数を用いた収量予測への応用

検出 AI を用いた花芽数や果実数の推定が可能であることが示されたので、先行研究 (Salvo et al. 2012) により示されている花芽数と果実数の相関を、検出 AI と画像データを用いて同様の相関関係が得られるか検証した。その結果、果樹ごとに画像から手作業で花芽と果実を測定した場合と、果樹部分を切り抜いた画像から AI を用いて推定した場合のどちらにおいても、花芽数と果実数の間に正の相関が見られた (図 7)。



(a) 手作業で求めた値



(b) v5s2560・2560 で得られた結果から回帰した値

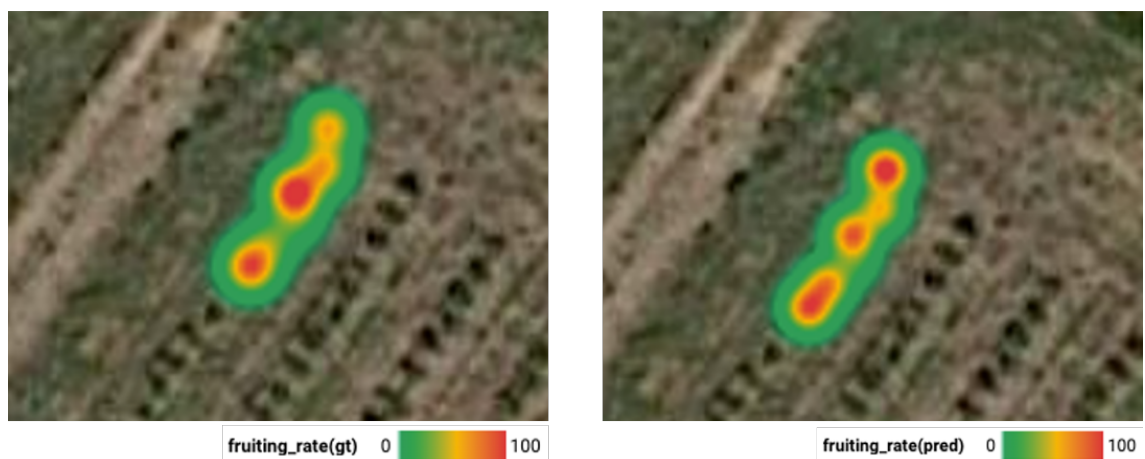
図 7 花芽数を用いて収量予測を行ったグラフ

この結果から検出 AI を用いて花芽数から果実数を推定することが可能であることが示された。検出 AI の導入によって、従来ではデータ収集のコストの問題により実現が困難であった、花芽数に基づく果実数の推定が行えることが期待される。しかし、興味深いことに画像中から手作業でのカウントを行った場合には、検出 AI を用いた方法よりも相関係数が高い値を示した。このことから、検出 AI を用いて花芽数から果実数を推定する際には回帰誤差だけでなく、隣接する果樹間の重なりやその他の外部要因により、正確な数値を計測することが難しい場合があると考えられる。今後、検出 AI をさらに洗練させ、花芽数を圃場全域で推定する手法を確立することで、ブルーベリーの収穫量を事前により正

確に予測することが可能となると考えられる。

2.3.4 結実率の可視化

蜂の巣箱の効果的な配置や収量減少の発生箇所の特特定への利用が期待される、結実率の可視化を行った。手作業および検出 AI によって求めた花芽数と果実数から算出される結実率を Google マップ上に表示させた（図 8）。ただし、ヒートマップの色合いを強調するために結実率の値は 20 から 100 の値に正規化した。



(a) 手作業で求めた結実率

(b) v5s2560・2560 で得られた結果
から回帰した結実率

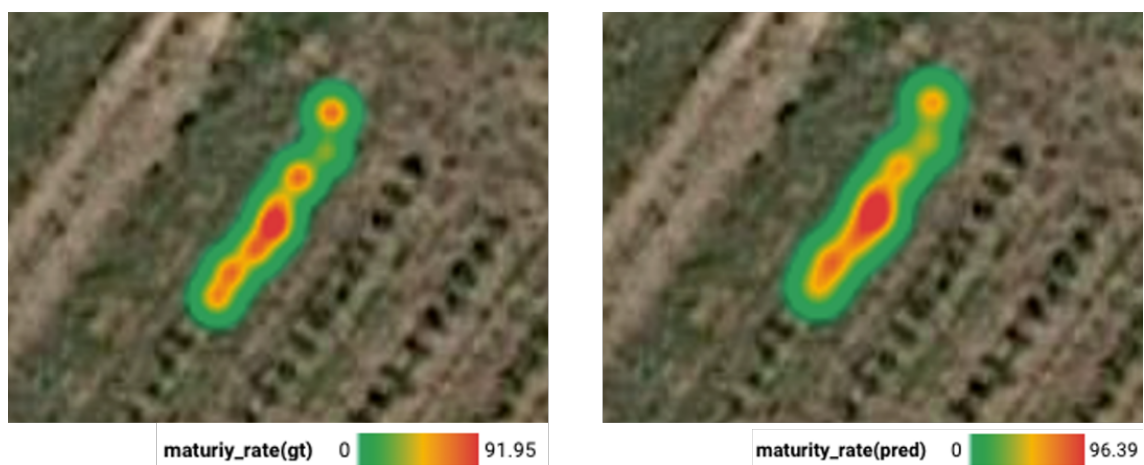
図 8 結実率のヒートマップ

ヒートマップの強度は検出 AI による推定誤差により両者で異なる点が見られたが、概ね同様の結実率の分布が見られ、検出 AI による結実率の可視化を実現

した。今後、回帰精度を向上させた高精度な検出 AI を用いて圃場センシングの自動化を行うことで、圃場全域で結実率の分布の可視化が可能になり、蜂の巣箱の位置などが与える結実率への影響を定量的に判断することが可能になる。また、結実率が低い部分の特定が可能となり、収量増加のためのデータへの利用が期待される。

2.3.5 成熟果実率の分布および成熟果実率の推移の可視化

適切な収穫場所や収穫時期のために利用が期待される成熟果実率のマップ化を結実率のヒートマップの場合と同様に行った（図 9）。その結果、ヒートマップの強度は検出 AI による推定誤差により両者でわずかに異なるが、同様の成熟果実率の分布が見られ検出 AI による成熟果実率分布の可視化が実現できた。



(a) 手作業で求めた成熟果実率

(b) v5s2560・2560 で得られた結

果から回帰した成熟果実率

図 9 成熟果実率のヒートマップ

さらに、同一果樹を定期的に撮影することで、果実数や成熟果実率の時系列データを取得することができる。そこで一本の果樹を対象にして画像から果実数と成熟果実率を手作業および、検出 AI による推定から取得し、それぞれを時系列順にプロットした (図 10)。

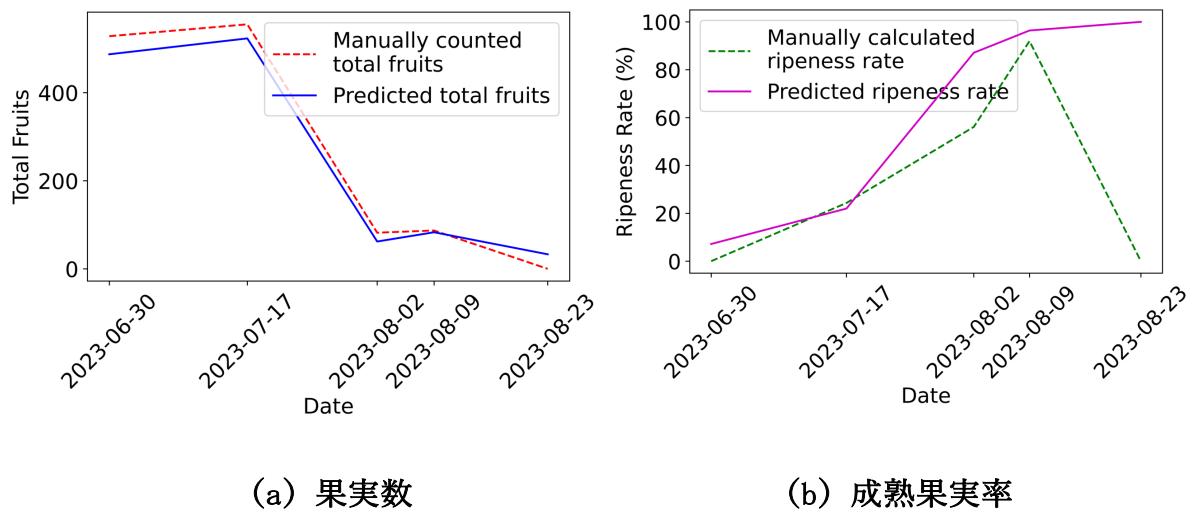


図 10 果実数および成熟果実率の推移

検出 AI による推定誤差を含んでいるため両者は完全に一致はしていないものの、7 月 17 日に以降に収穫が行われたことや、収穫後期になるにつれて成熟果実率が増加することが可視化されたことがグラフから読み取れ、果実数および成熟果実率の時系列データを可視化することができた。ただし、成熟果実数を求める回帰式の切片が正の値を示しており、検出 AI によるカウントにおいて成熟と未成熟の果実数がどちらも 0 になった場合に成熟果実率が 100%を示してしまったため、8 月 23 日の成熟果実率が大きく異なった。果実数が少なくなった際の回帰式については検討する必要がある。

本研究で作成した時系列データを圃場全域で集計し、複数年のデータを蓄積することで成熟果実率のピーク時期を予測することが可能になり、最適な収穫時期の決定のためのデータとしての利用が期待される。

2.4 結論

この章では物体検出 AI を用いたブルーベリーの花芽と果実の計数および分類の有用性を評価した。まず、YOLOv5 と YOLOv8 を使用し、ブルーベリーの花芽数と果実数を推定した結果、凍害を受けた花芽に関しては改善の余地があるものの、一つの果樹が収まる画像から花芽と果実の検出・分類が可能であることが示された。また、640 ピクセルの低解像度画像を用いた推論においても高い決定係数を示した。単回帰式を用いた画像中の花芽と果実の計数においては、過学習の問題が生じたものの MAPE の最小値は花芽で 10.79%、果実で 23.84% となった。一方で、物体検出 AI の適用には画像中の果樹を手動でトリミングする必要があったため、今後の課題として例えば物体検出 AI を用いて個々の果樹を自動でトリミングするなどの手法開発が求められる。この手法については第四章で検討を行った。

第 3 章 画像を用いた果実の推定についての検証

3.1 概要

画像を用いて果樹の総果実数を推定する際に、結実期の葉によって一部の果実が隠れてしまうことが果実数推定に悪影響を及ぼす可能性が懸念される。そこでこの章では、こうした問題に対処するため果樹の葉の繁茂度合いや撮影角度が推定精度に与える影響を検討した。

3.2 材料と方法

3.2.1 画像の撮影及び真値の取得方法

画像に映る果実数と実際の果実数に相関があるかを調べるために、物体検出 AI 開発で対象としたのと同じブルークロップ品種を対象に果樹ごとの実際の果実数を計測した。14 本の果樹を対象に二名で計測を行い、総果実数は 10891 個となった。計測した果樹を樹列の両側（北西方向および南東方向）から撮影し、計 28 枚の画像データを取得した（図 11）。果実の計測および撮影は収穫時期以前である 6 月下旬に実施し、全ての果実が未成熟であった。第二章で開発した Android アプリケーションおよび `labellmg` を用いて果実のラベリングを行い、バウンディングボックス数を画像に映る果実数とし、画像に映る果実数と実際に計測した果実数のピアソンの相関係数を算出した。また、画像に映る果実数から実際に計測した果実数を求める単回帰式を作成し、その予測値と実際に計測

した果実数の誤差を MAPE により算出した。

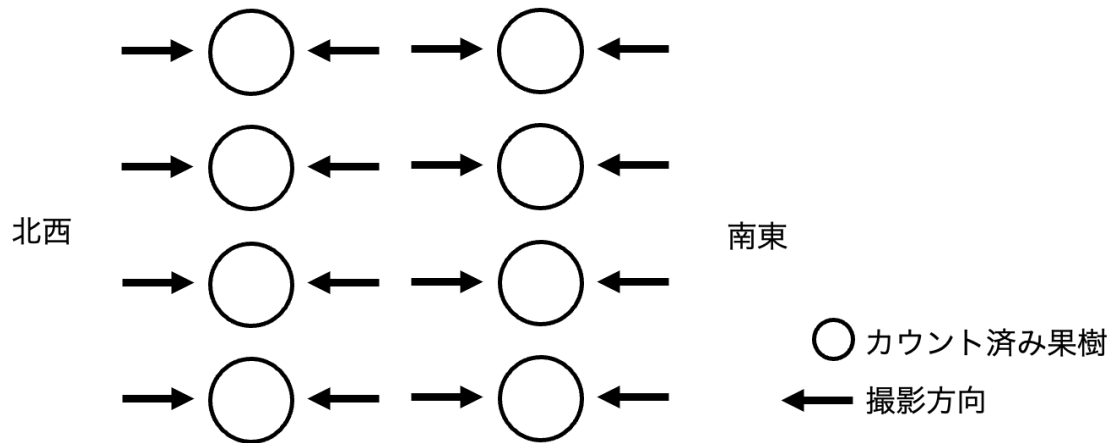


図 11 果実数を計数した果樹の撮影方法

3.2.2 葉の被覆率による果実数推定への影響の検証

果樹の葉の生い茂り方により、画像中の果樹における葉の被覆率が異なるため、葉が少ないほど多くの果実が映り込み、葉が多いほど画像に映る果実の割合が減少すると考えられる。画像に映る果実の割合が変化することで、実際の果実数を推定する際の誤差要因になると考えられるため、葉の被覆率が与える画像に映る果実数の割合および、果実数の推定結果に与える影響を検証した。

葉の被覆率を求めるために、葉や果樹をポリゴンで囲う必要がある。画像中にポリゴンを描画するための既存のソフトウェアは PC 上で実装されており、マウスで操作する必要がある。しかし、本研究で撮影した果樹は葉が多く、複雑に分布しているため、マウスを使ってポリゴンを作成するには非常に時間がかか

る作業である。

そこで本研究では、ポリゴンの描画作業を高速化するため、タブレット上でポリゴンをタッチペンで描画を可能にする、独自の Android アプリケーションを開発した（図 12）。このアプリケーションは第二章で開発した Android アプリケーションをもとに開発を行った。



図 12 被覆率算出のための Android アプリケーション

開発言語は Kotlin を使用し、OpenCV (Open Source Computer Vision Library) と呼ばれる動画や画像を処理するためのオープンソースライブラリを使用した。作成したアプリケーションは、タッチペンでポリゴンを描画し、作成したポリゴンに白黒画像として保存することが可能で、葉の部分をマスキングした白黒画

像と、果樹の部分をマスキングした白黒画像の、それぞれのピクセル数の比をとることで被覆率を求めることが可能である。タッチペンや指で直感的に画像の拡大や移動などの操作や、ポリゴンが描画可能であることに加え、OpenCV による高速な画像処理を実現していること、元画像とポリゴンを描画した画像を別々に保存する設計を採用していることで、ポリゴンの表示及び非表示も高速で行うことができることから、開発したアプリケーションはポリゴンを描画する上で利便性の高いものとなっている。このアプリケーションを利用して果樹および葉のポリゴン画像を作成し、撮影した果樹ごと葉の被覆率を算出した（図13）。ただし、果樹については、枝を除外した果樹の外周をなぞるようにマスキングを行った。



(a) 元画像



(b) 葉のマスク画像



(c) 果樹のマスク画像

図 13 被覆率の算出過程の例

画像を用いて実際の果実数を推定する際に、被覆率が与える影響を以下の手順により検証した。まず、実際の果実数に対する画像に映る果実数の比率を計算し、同時にその時の被覆率を求めた。被覆率と実際の果実数に対する画像に映る果実数の比率のピアソンの相関係数を算出した。次に、画像に映る果実数から実際の果実数を推定する回帰式を単回帰分析により作成した。果樹ごとに実際の果実数から回帰式による予測値を引いた残差 Δf と残差の絶対値 $|\Delta f|$ を求めた。

先述の葉の被覆率と実際の果実数に対する画像に映る果実数の比率のデータを参照し、被覆率が低く果実の露出が顕著な 6 枚の画像とそれ以外の画像で群分けを行った。各群の Δf および $|\Delta f|$ それぞれについて Mann-Whitney の U 検定を用いて統計解析を行い、有意水準を 0.05 として統計的有意差があるか検証した。

3.2.3 撮影方角および両側撮影による果実数推定への影響の検証

ブルーベリー果樹は列状に植えられており、果樹は樹列を挟んで両側から撮影が可能である。果樹の撮影方角によって日当たりの違いなどから葉の繁茂度合いが異なり、同一果樹においても画像に映る果実数に影響することが考えられるため、その検証を行った。まず、撮影方角が果実数の推定結果に与える誤差の影響を検証した。被覆率の場合と同様に北西方向の群と南東方向の群に Δf および $|\Delta f|$ を分けて、それぞれの群について Mann-Whitney の U 検定を用いて統計解析を行い、有意水準を 0.05 として統計的有意差があるか検証した。また、画像中に映る果実数と実際の果実数のピアソンの相関係数を撮影方角別に算出し、同じ方角から撮影した画像のみを用いて実際の果実数が推定可能であるかについて検証した。

次に、樹列の両側の画像を用いて実際の果実数を推定した場合、樹列の片側の

みの場合と比べてより高精度な推定が期待される。一方で両側の場合には 1 列の果樹に対して 2 回撮影を行う必要があるため、圃場モニタリングのコストが増加する。そこで、両側から果樹を撮影して 2 枚の画像に映る果実数の平均値から実際の果実数を推定する場合と、片側のみの画像を用いて実際の果実数を推定する場合の差異について検証した。被覆率や撮影方角の検証と同様に、片側の画像に映る果実数から実際の果実数を推定する回帰式と、両側の画像に映る果実数の平均値から実際の果実数を推定する回帰式を単回帰分析により作成した。果樹ごとに実際の果実数から、回帰式による予測値を引いた残差の絶対値 $|\Delta f|$ をそれぞれ求めた。各群における $|\Delta f|$ について Mann-Whitney の U 検定を用いて統計解析を行い、有意水準を 0.05 として統計的有意差があるか検証した。

3.3 結果と考察

3.3.1 画像に映る果実数と実際の個数との関係

ブルーベリーの果樹を撮影した画像を用いて AI による果実数の推定が実際の果実数を反映しているか確かめた。その結果、画像に映る果実の個数と実際に果樹に結実している果実数との間には正の相関関係が見られた (図 14)。また、単回帰式から得られた予測値と真値との誤差は MAPE で 21.2% となった。

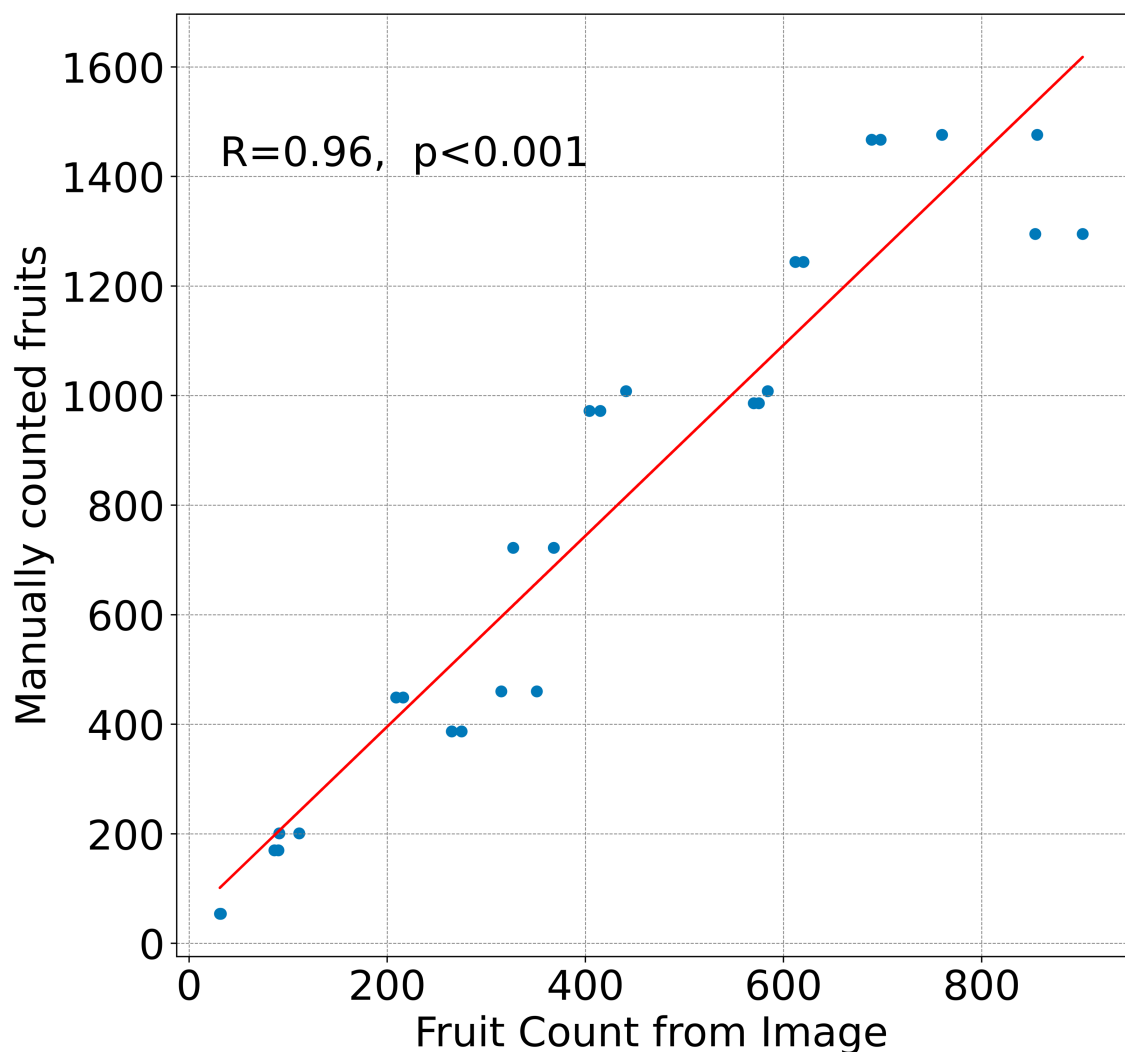


図 14 画像に映る果実数と実際の果実数の関係

このことから、一部の果実が葉に隠れており全ての果実が画像に写っていない状況でも画像を用いて実際の果実数の推定は可能であることが示された。つまり画像に映る果実数を高精度に推定できる果実検出 AI を使用することで、実際に果樹に結実している果実数を推定が可能であることが示唆された。また、物体検出 AI で計数した果実数を用いてローブッシュブルーベリーの収量推定を行

った先行研究 (MacEachern et al. 2022) では、収量の平均値に対する予測値と真値の平均絶対差の比率により計算された誤差で 24.1% となり、同程度の果実数の推定結果が得られた。ただし、果樹が密集して生育される樹列においては一枚の画像上に複数の果樹が写り込むことが予想される。このような場合に単一果樹ごとの果実数を正確に推定するためには、単一の果樹を画像中から正確にトリミングする方法の開発が求められる。これにより果樹が密集している樹列においても単一果樹ごとに果実数を正確に推定できると考えられる。

3.3.2 葉の被覆率による果実数推定への影響の検証

果樹における葉の被覆率が高くなると果実が葉に隠れることで、画像に映る果実数が減少すると考えられる。そこで、画像に写った果実数と実際の果実数の比が被覆率によって変化するかを検証した。その結果、被覆率が上昇すると画像に映る果実数が減少することが示された (図 15)

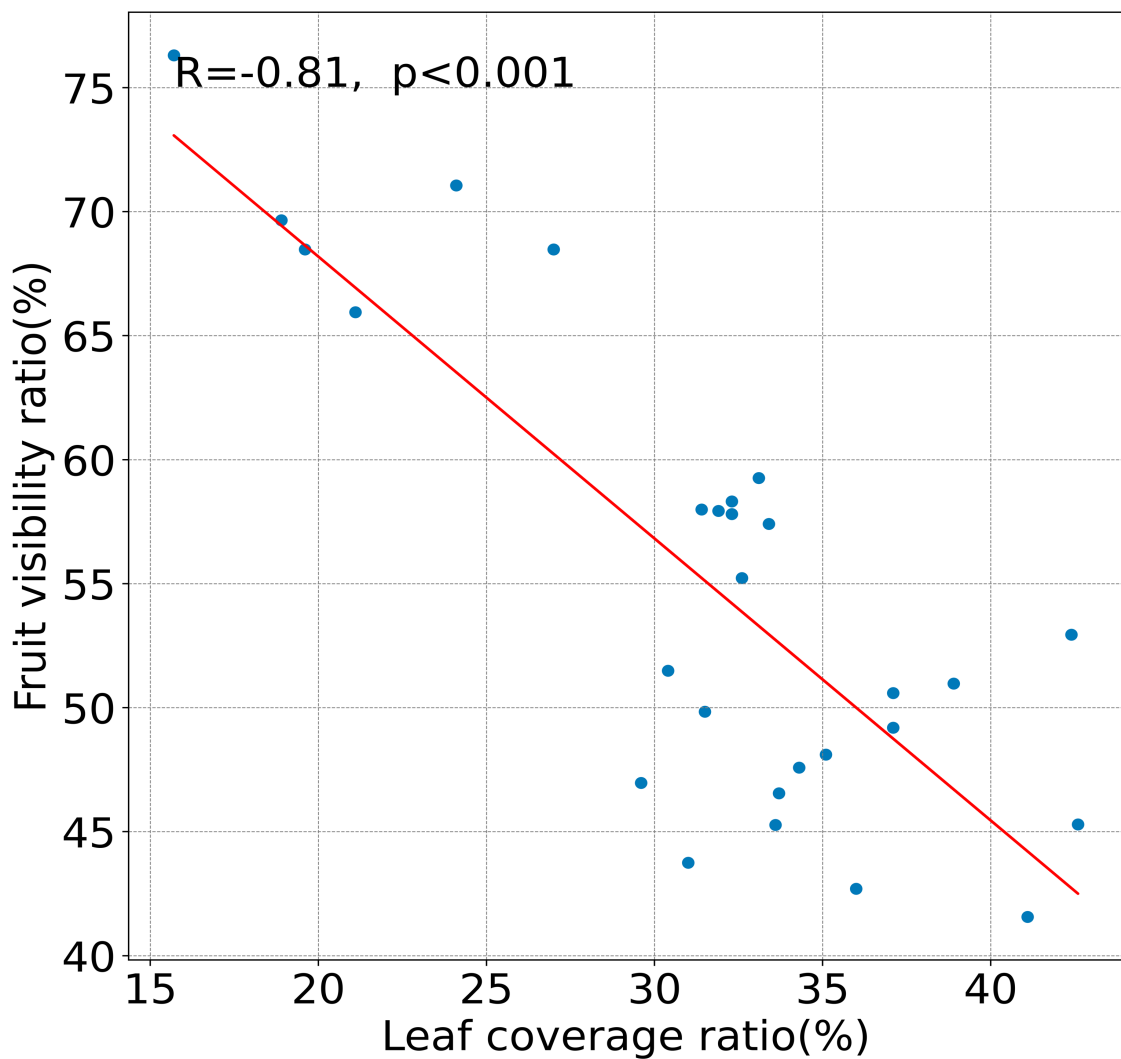


図 15 実際の果実数に対する画像に映る果実数の比率と被覆率の関係

そこで、画像から実際の果実数を推定する際に被覆率が与える影響を検証した。図 15 のデータの分布から果実の露出が顕著で被覆率が低い 6 枚の画像と、それ以外の画像の 2 群に分類し、それぞれのデータ群を被覆率の低い群と高い群とした。次に、単回帰分析により全データから実際の果実数を推定する回帰式を算出した (図 16)。

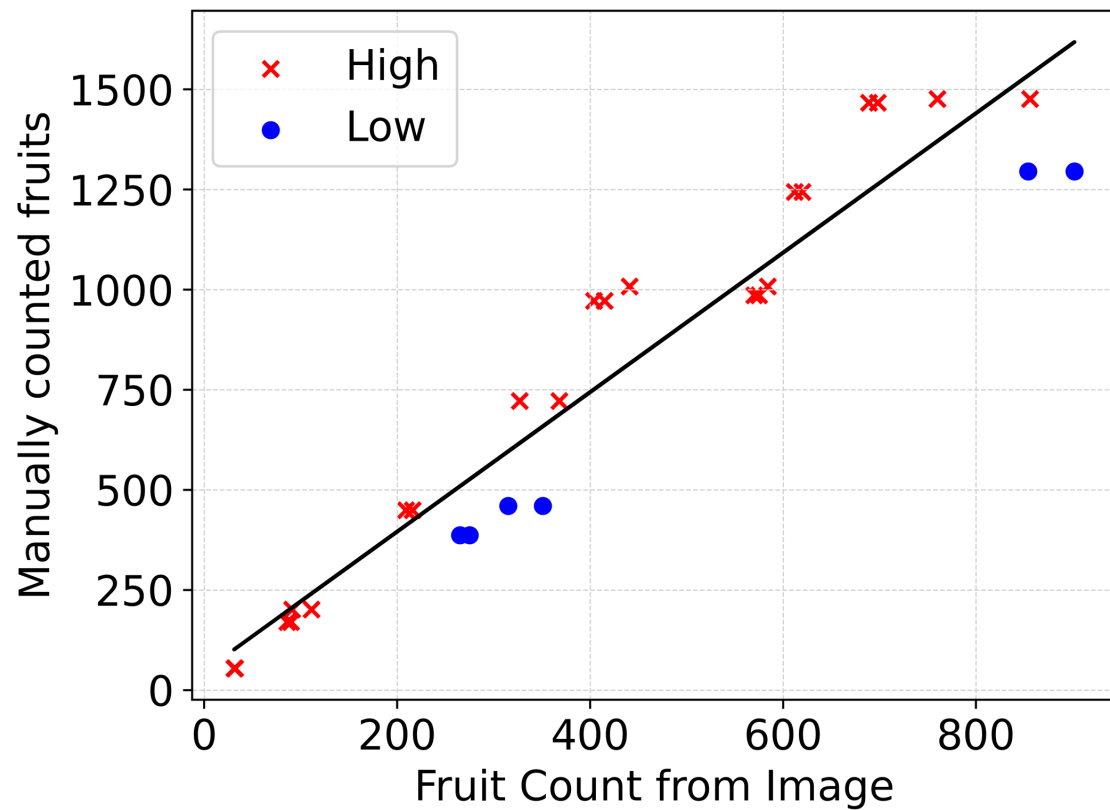


図 16 被覆率ごとに群分けした、画像に映る果実数と実際の果実数の関係

それぞれの群における各サンプルの実際の果実数から予測値を引いた残差 Δf および、その絶対値 $|\Delta f|$ を算出した。被覆率が高い群の実際の果実数は回帰直線よりも大きい傾向が見られ、被覆率が低い群の実際の果実数は回帰直線よりも小さい傾向が見られた。次に、それぞれの群における各サンプルの実際の果実数から予測値を引いた残差 Δf および、その絶対値 $|\Delta f|$ を算出した (図 17)。

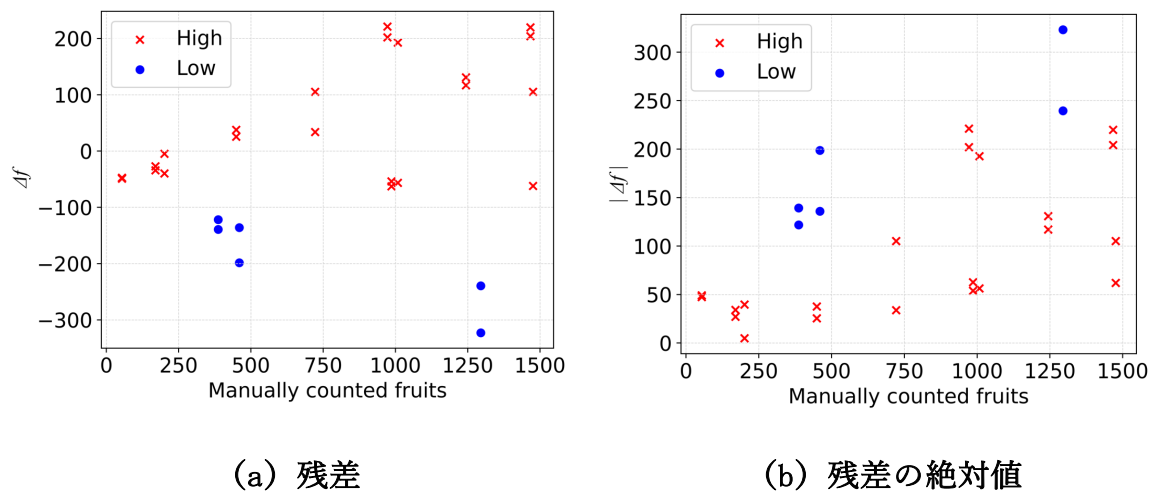


図 17 被覆率ごとに群分けした、実際の果実数と残差との関係および、残差の絶対値との関係

その結果、 Δf の値は被覆率の高い群が低い群よりも有意に高くなり、 $|\Delta f|$ の値は被覆率の高い群が低い群よりも有意に低くなった (図 18)。この結果から、被覆率が誤差に影響を与えることが明らかになった。

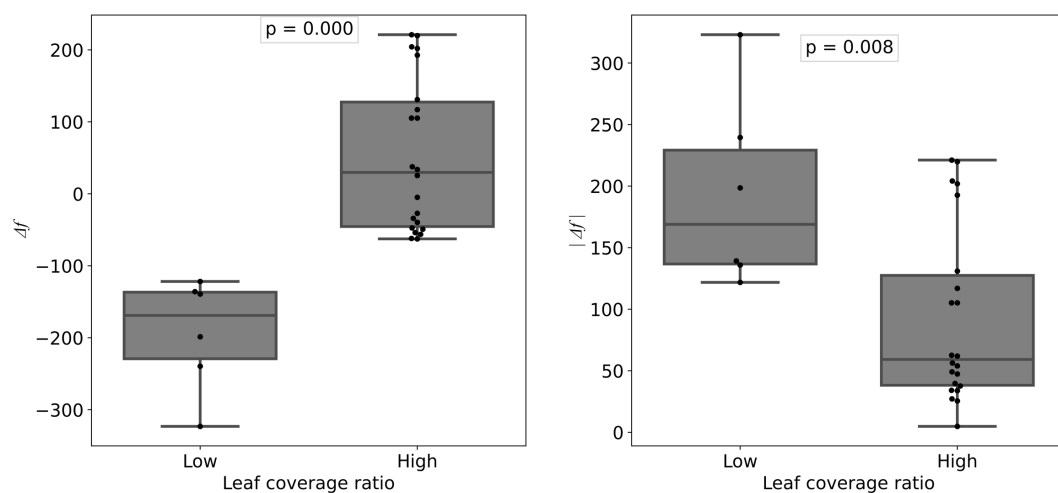


図 18 被覆率で群分けした、残差および残差の絶対値のグラフ

次に、画像から実際の果実数を推定する精度に、被覆率が影響することが判明したため、被覆率の高さに応じて異なる回帰式を単回帰分析により求めた。その結果、被覆率が高い場合と低い場合では異なる回帰式が算出された（図 19）。被覆率が高い場合は予測値が大きくなる傾向にあり、被覆率が低い場合は予測値が小さくなる傾向にあることが示された。

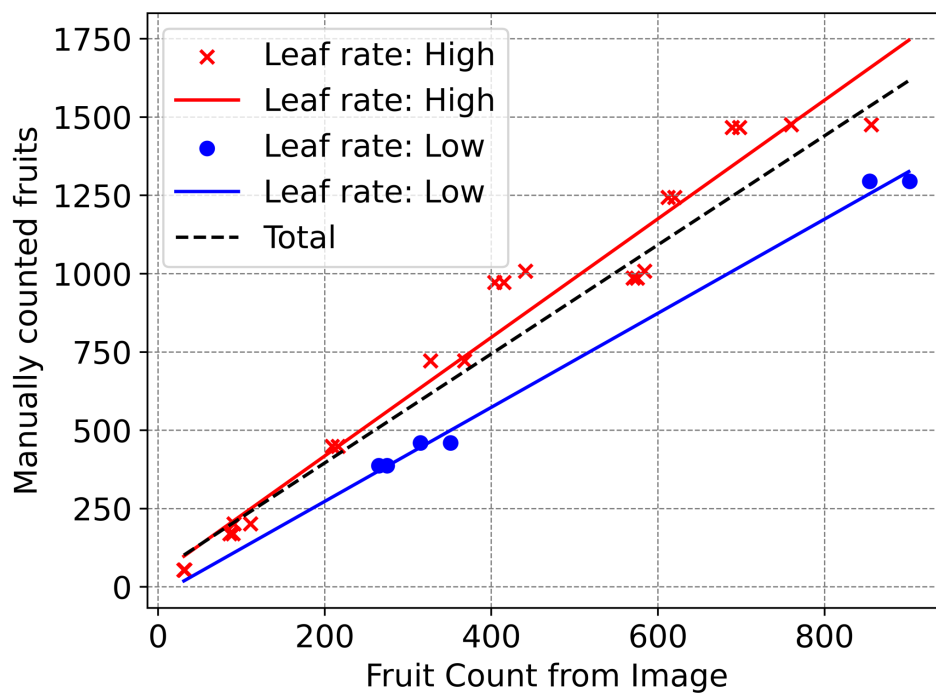


図 19 被覆率ごとに計算した回帰直線

以上の結果から葉に隠れた果実を含む実際の果実数を推定する際は、葉の被覆率を考慮することが有効と考えられる。また、被覆率を考慮して推定モデルを切り替えることでさらなる推定精度向上が期待できる。ただし、本研究では画像

に映る果実数と実際の果実数の関係についてのデータ数に限りがあった。今後データ数を増やし、さらなる検証を行うことで被覆率が果実数推定に与える、より詳細な関係を明らかにし、被覆率を考慮した高精度な果実推定モデルの実現が期待される。また、本研究では被覆率を手作業によって求めたが、1枚の画像あたり1時間半ほどの時間がかかることから手作業による被覆率の算出は現実的ではない。よって今後、被覆率を求めるためにU-Net(Ronneberger et al. 2015)などのセマンティックセグメンテーションAIを用いた被覆率の自動推定を実現することで、被覆率を考慮した精度の高い果実数推定が期待される。

3.3.3 撮影方角および両側撮影による果実数推定への影響 の検証

本研究で対象とした圃場は南北方向に対して約45度傾いた列状に果樹が植えられており、日当たりの違いなどによって葉の繁茂度合いが異なる可能性を考慮し、果樹の撮影方角および撮影方法と果実数推定の関連性について検証した。まず撮影方角が果実数推定に影響を与えるかについて検証した。被覆率の場合と同様に、撮影方角に応じて群分けを行い、単回帰分析によって全データから実際の果実数を推定する回帰式を算出した(図20)。

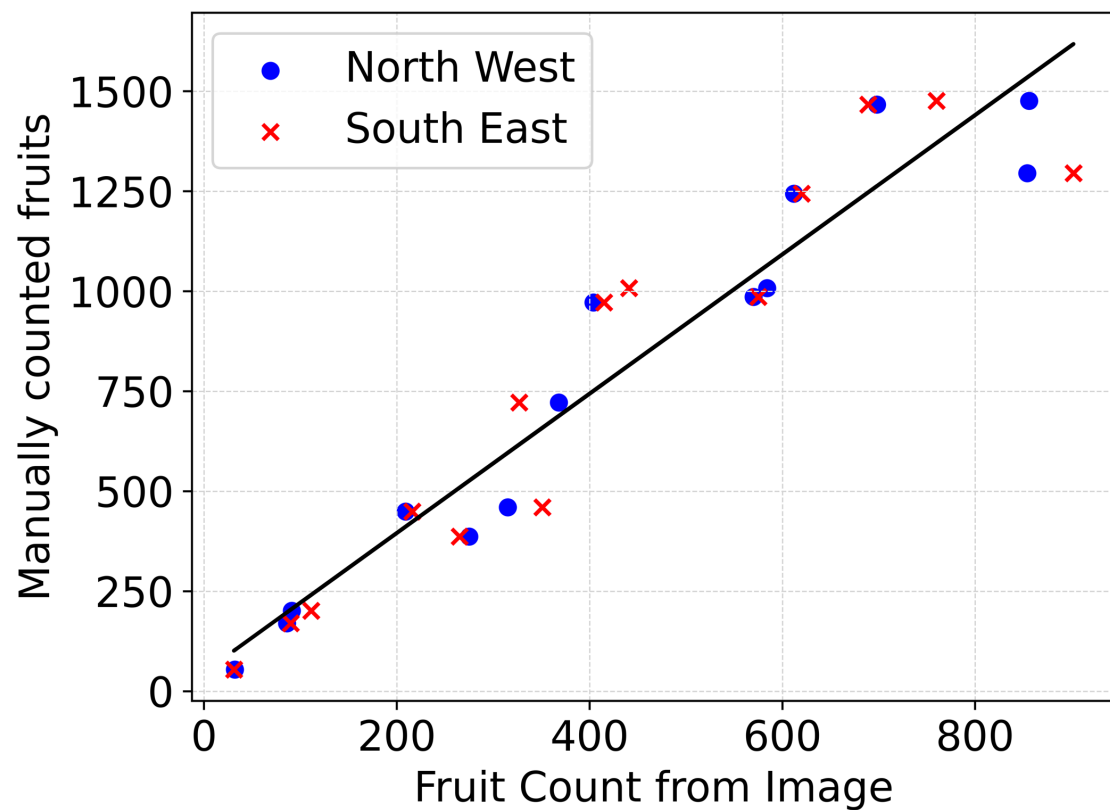


図 20 撮影方角ごとに群分けした、画像に映る果実数と実際の果実数の関係

各群による分布についての差異は確認できなかった。次に、それぞれの群における各サンプルの実際の果実数から予測値を引いた残差 Δf および、その絶対値 $|\Delta f|$ を算出した (図 21)。

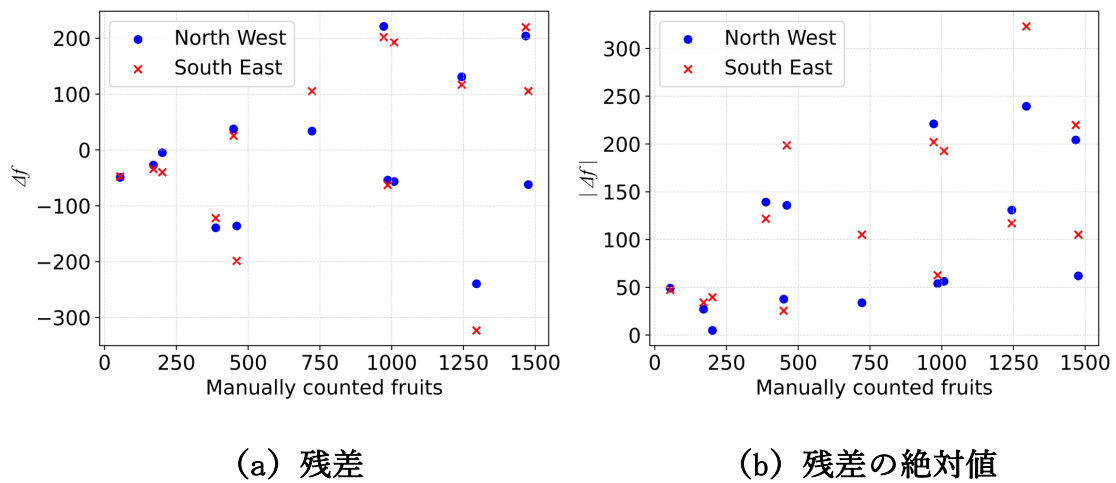


図 21 撮影方角ごとに群分けした、実際の果実数と残差との関係および、残差の絶対値との関係

その結果、撮影方角で分類した群には Δf の値および、 $|\Delta f|$ の値には有意な差は見られなかった (図 22)。

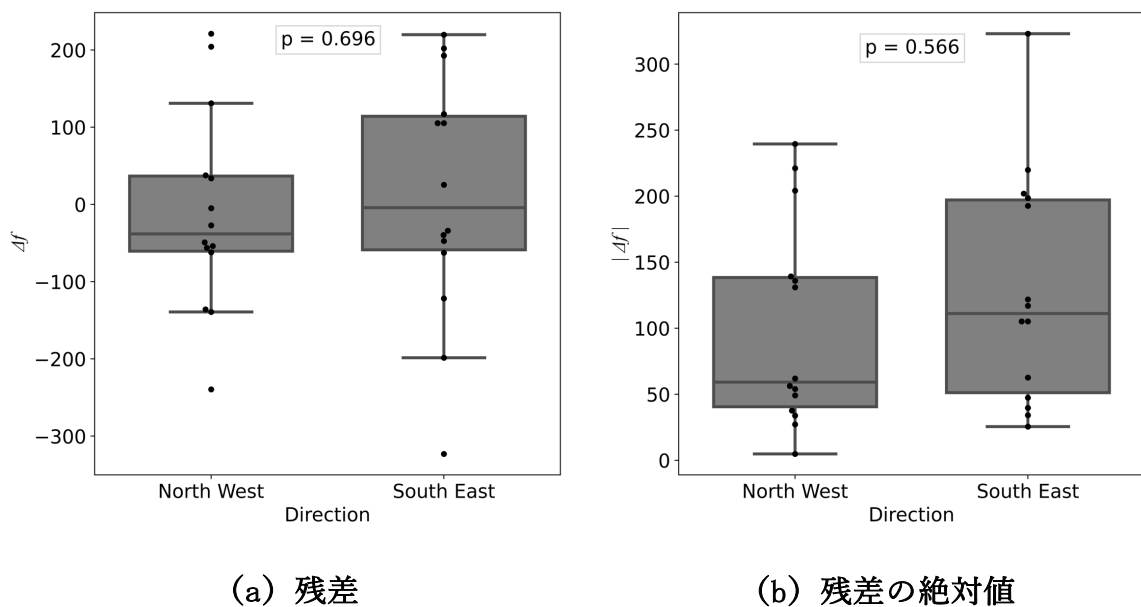


図 22 撮影方角ごとに群分けした、残差および残差の絶対値のグラフ

また、同じ方角から撮影した画像を用いて、実際の果実数が推定可能であるかを確かめた。その結果、北西方向から撮影した場合と南東方向から撮影した場合のどちらにおいても、画像に映る果実数および実際の果実数には正の相関関係が見られた（図 23）。

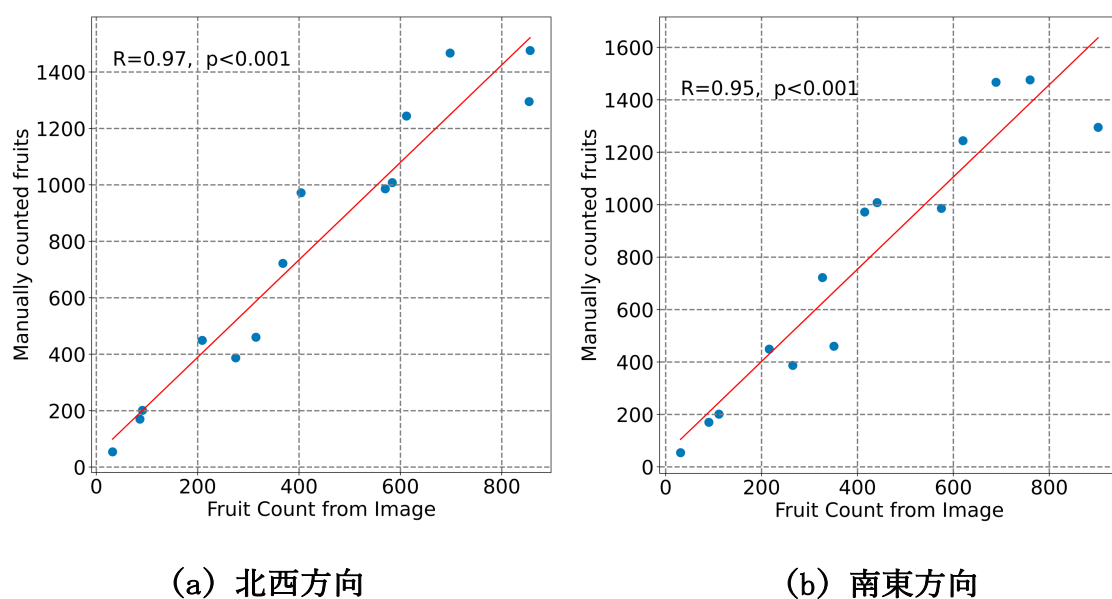


図 23 撮影方角別の画像に映る果実数と実際の果実数の関係

さらに、画像から果実数を推定する際に、樹列を挟んで両側から撮影した果樹の画像を使用する場合と、片側の画像のみを使用する場合に推定精度に差異が生じることが予想されたため、実際の果実数推定における両側撮影の必要性について検証した。

まず、片側の画像に映る果実数から実際の果実数を推定する回帰式と、両側の

画像に映る果実数の平均値から実際の果実数を推定する回帰式を単回帰分析により、それぞれ算出した。次に、それぞれの群における各サンプルの実際の果実数から予測値を引いた残差の絶対値 $|\Delta f|$ を算出した（図 24）。

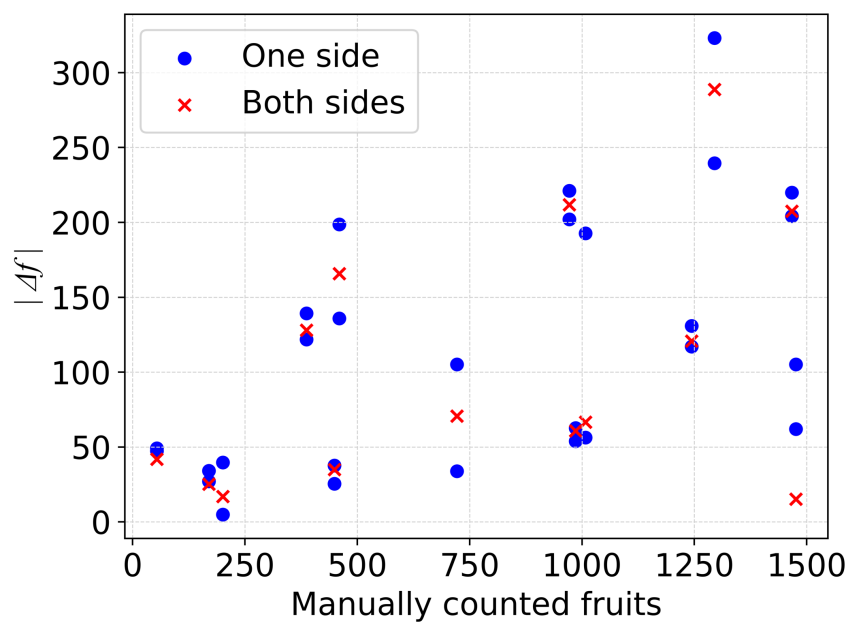


図 24 片側撮影および両側撮影で群分けした、実際の果実数と残差との関係
その結果、樹列の片側の画像のみを使用した場合と両側の画像を使用した場合に $|\Delta f|$ の値には有意な差は見られなかった（図 25）。

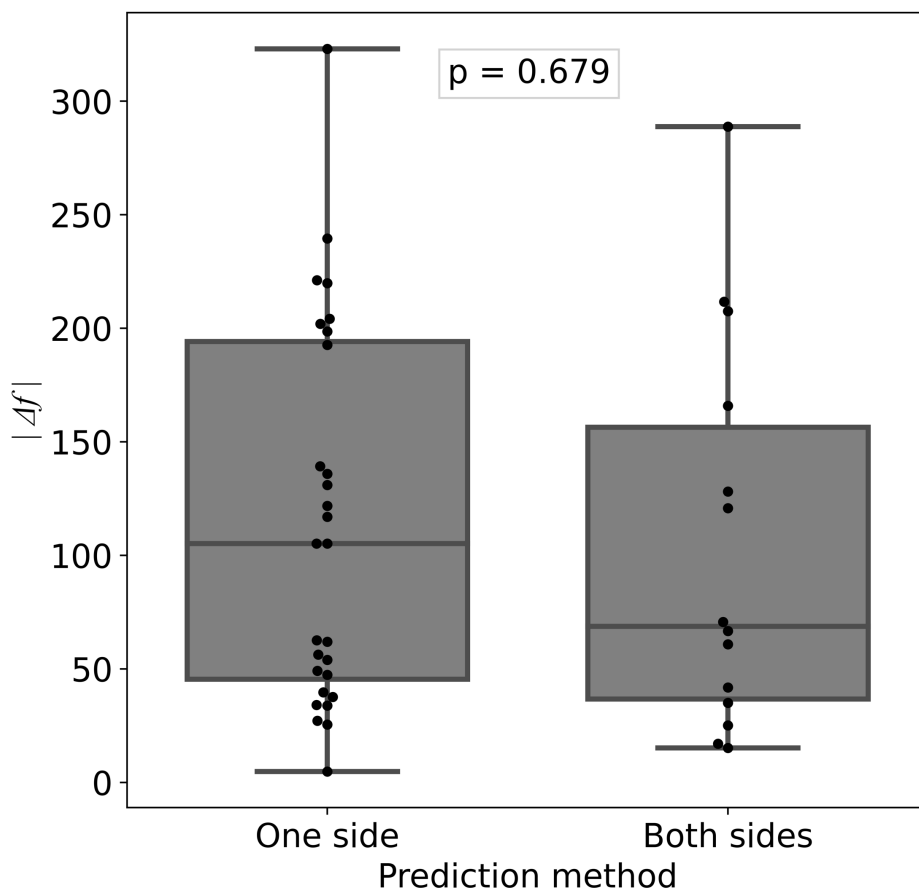


図 25 片側撮影および両側撮影で群分けした、残差の絶対値のグラフ

以上の結果から本研究で対象としたブルーベリー果樹においては実際の果実数の推定誤差には、果樹の撮影方角が影響しないことが明らかとなった。また、先行研究でオレンジの果樹では複数枚の画像を使用した場合に、実際の果実数の推定精度が向上することが示されていたが (Mirhajia et al. 2021)、本研究では樹列の両側から撮影する必要があることが示された。すなわち、果実の自動計数を実用化する際は、任意の作業道から果樹を 1 回のみ撮影すれば良いことが

明らかとなった。

3.4 結論

この章では結実期のブルーベリー果樹を撮影した画像では葉に隠れてしまう果実があることを考慮し、画像中の果実数から実際の果実数が推定可能かどうかについて検証した。その結果、画像に映る果実数と実際の果実数には正の相関 ($R=0.96$, $p<0.001$) があることを明らかにした。また単回帰式との回帰誤差は MAPE で 21.2% となり、画像から果実数を推定することの妥当性を示した。

さらに、高精度かつ効率的な圃場モニタリングの自動化を実現するために、画像から果実数を推定する際に葉の被覆率、撮影方角、樹列の両側からの撮影が推定精度に及ぼす影響を解析した。その結果、葉の被覆率が推定精度に大きく影響することが明らかとなった。このことから今後、画像を用いて高精度に果実数を推定するためには、セマンティックセグメンテーション AI などによって算出された果樹の葉の被覆率の利用が有効であることが考えられる。一方で、撮影方角の考慮や樹列の両側から撮影することは推定精度に影響がなく、画像を用いた果実計数の際は同一の果樹を樹列の右側と左側に関わらず、一度撮影するだけで十分である可能性が示された。これにより、撮影のための作業経路を短縮でき、圃場全体の効率的なモニタリングが実現可能であると思われる。

第 4 章 果樹トラッキングシステムの開発

4.1 概要

物体検出 AI を用いて花芽や果実の自動計数を行うためには、果樹ごとの画像を自動取得することが求められる。ブルーベリー圃場では果樹を列状に植えるのが一般的であり、果樹同士が隣接して重なり合う場合が多く存在する。そのため、1 枚の画像には複数の果樹が映り込むことがあり、画像から 1 本の果樹を認識し切り抜く必要がある。また、人手で静止画の撮影を行うのは作業量の観点で困難であるため、圃場モニタリングの自動化は作業車などで樹列間を走行し、取り付けたカメラで撮影された動画を解析することで実現されと考えられる。果樹の認識には物体検出 AI の利用が有用であると考えられるが、動画に物体検出を適用するためには、検出した果樹を追跡しフレームごとに紐づける必要がある。そこでこの章では樹列を動画で撮影し、撮影された動画から果樹ごとの画像を自動的に取得するシステムの開発を行った。

4.2 材料と方法

4.2.1 Robot Operation System (ROS)

本研究で開発したシステムのソフトウェアは Robot Operation System(以下 ROS)と呼ばれるロボット開発用プラットフォームを用いた (ROS.org 2023)。ROS とは主にロボット産業で広く用いられるプラットフォームであり、ロボッ

ト開発に有用なライブラリや可視化ツールなどを含んだフレームワークである。

ROS を用いてソフトウェアを開発することで、分散型プログラムの作成を容易に行うことができる。ROS を用いたソフトウェアでは、プログラム全体をノードと呼ばれる小さいプログラムに分割し、ノードごとに様々なタスクを割り当てて運用を行う。ノード間ではメッセージと呼ばれる形式で画像などのセンサデータやロボットへの命令データなどを送受信する。メッセージの送受信には配信者 (Publisher) と購読者 (Subscriber) の形式を採用している。データを送信する際は、ROS CORE と呼ばれるマスターにメッセージ名を指定して配信し、データを受信する際は ROS CORE に配信されているメッセージを購読することでデータの送受信が可能になる。この機能により、容易にデータの送受信が可能になるため分散システムの作成が容易になる。

また、ROS を使う利点としてセンサデータを bag ファイルと呼ばれる形式で保存し、オフラインでシミュレーションが可能である点である。そのため、現場で取得した bag ファイルを用いて後日シミュレーションを行い、データの解析が可能になる。特に農業現場では PC 環境などが制限されることが想定されるため、センサデータのみ取得し、後日解析するといった使い方が可能である。本研究においても圃場で取得したセンサデータを bag ファイル形式で保存し、データの解析をオフラインで行った。

4.2.2 データ取得システム

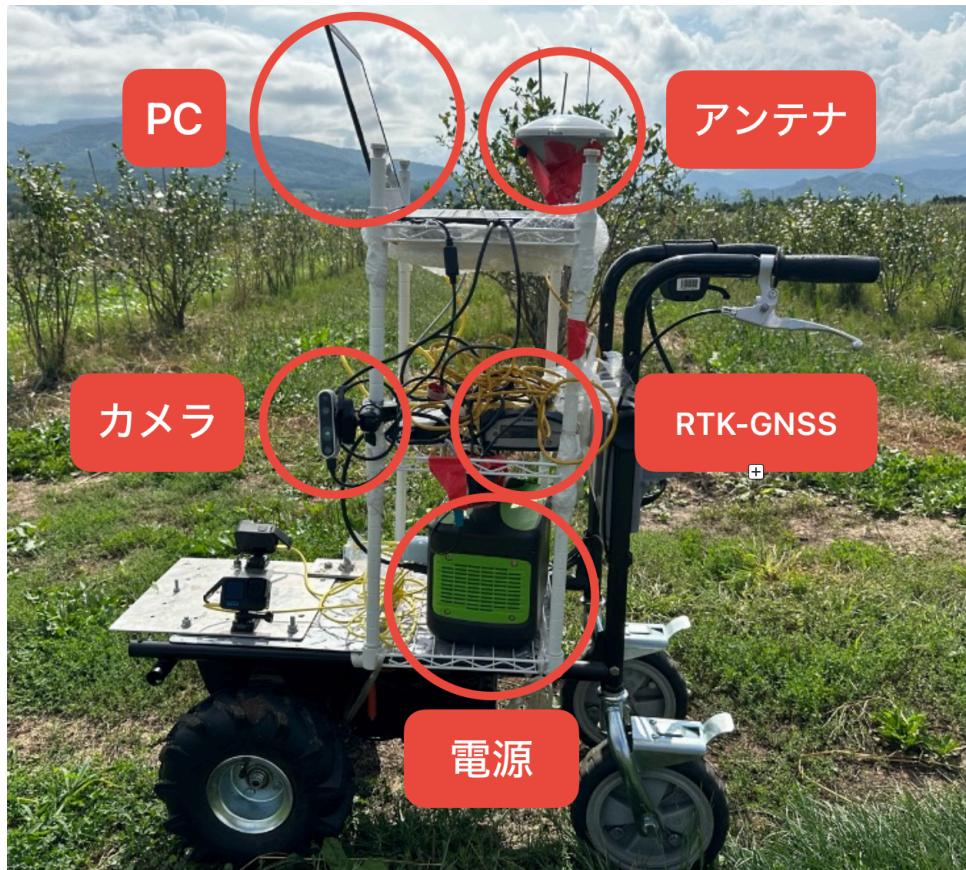


図 26 圃場見回りシステム

圃場内の動画データおよび位置データを取得するために、小型電動運搬台車に RTK-GNSS、GNSS 用のアンテナ、カメラ、データ取得用の PC、ポータブル電源を取り付けたデータ取得システムを作成した (図 26)。台車にはアクセルがついており、アクセルを押しながら手で方向を制御することで、圃場内を自由に走行させることが可能である。PC には ROS が組み込まれており、カメラと RTK-GNSS から送られるデータを bag ファイル形式で保存することが可能である。

RTK-GNSS は SPS855 (Trimble 社製) (図 27)、カメラは RealSense D435i (Intel

社製）（図 28）を用いた。



図 27 使用した RTK-GNSS



図 28 使用したカメラ

4.2.3 果樹検出 AI の作成

複数の果樹が映る静止画から果樹ごとの画像を取得するためには、画像中から果樹部分を検出する必要がある。そこで本研究では物体検出 AI を用いて単一果樹の画像を取得した。物体検出 AI による果樹検出モデル作成のために YOLOv5 を使用した。モデル学習のデータセットにはデータ取得システムにより撮影された動画から取得した静止画を使用し、果樹のアノテーションを行った。学習用データとして 351 枚アノテーションを行い、そのうち 80%を学習用、20%を検証用データとして使用した。また、学習データ数を増やすためにデータ拡張を行った。画質を 50%と 75%に落とした画像、明るさを 3 段階に変化させた画像を追加し、元の画像と追加した画像を上下反転させることで、学習データを 12 倍に拡張した。本研究では精度評価のためのデータを 2 種類用意した。一つは学習に使用した動画から取得した静止画に別途 92 枚アノテーションを行ったデータで、これをテストデータとして定義する。二つ目は学習データとは別の日に撮影された画像を使用し、50 枚アノテーションを行ったデータで、これを未知データとして定義する。学習に使用した PC には GPU として GeForce RTX3090 (NVIDIA 社)、CPU として Intel Core i9-12900K (Intel 社) を搭載し、Ubuntu (20.04) 上において Python(3.8.10)および PyTorch(1.13.1+cu117)を用いてモデルの学習を行った。学習時の最適化アルゴリズムには SGD を使用

した。また、エポック数を 200 に設定し、30 エポックごとに早期学習停止の監視を行った。バッチサイズについては 16 に設定した。YOLOv5 は学習時の画像解像度を 32 の倍数に設定することが可能であるため、640、1280 ピクセルに設定し学習を行った。モデルパラメータの初期値として YOLOv5s の公式事前学習済みモデルをベースとした。以上の条件により二つの果樹検出 AI モデルを作成した。

4.2.4 トラッキングアルゴリズム

動画に対して物体検出を適用し、果樹ごとに検出を行うためにはフレーム間で同一果樹のトラッキングを行う必要がある。そこで本研究では果樹のトラッキングアルゴリズムを開発した (図 29)。トラッキングにはカルマンフィルタを用いた。カルマンフィルタとは観測値を用いて動的システムの状態を推定するためのアルゴリズムである。本研究では果樹の画像中での検出位置を観測値として、次のフレームに映る果樹の画像中座標を推定するためにカルマンフィルタを使用した。本研究では Python のクラスを用いて Tracker というクラス名でカルマンフィルタを実装しており、トラッキングの際は果樹一つに対して一つのインスタンス化した Tracker を割り当て、Tracker を用いて観測値の更新や予測位置の取得を行う。果樹のトラッキングアルゴリズムの流れ図を図 29 に示す。まず、カメラから取得した画像に対して果樹検出モデルを適用し、画像中の果樹の

バウンディングボックスを取得する。次に Tracker が無い場合、すなわち果樹のトラッキングを行っていない場合はバウンディングボックスにそれぞれ Tracker を割り当てる。Tracker がある場合、すなわち果樹のトラッキングを行っている場合は Tracker とバウンディングボックス中心との距離を全て算出し、各 Tracker に最も近いバウンディングボックスの中心座標を観測値として各 Tracker を更新する。その後、Tracker は画像の左側に設定した直線を超えた場合と、トラッキングに連続で失敗した場合に削除する。この削除過程後に検出数より Tracker 数が多い場合は Tracker とバウンディングボックスの距離が近いペアをそれぞれ算出し、余った Tracker を削除する。その後 Tracker の ID を使ってバウンディングボックスにトラッキング ID を割り当てる。以上の過程によってバウンディングボックスのトラッキングを行う。

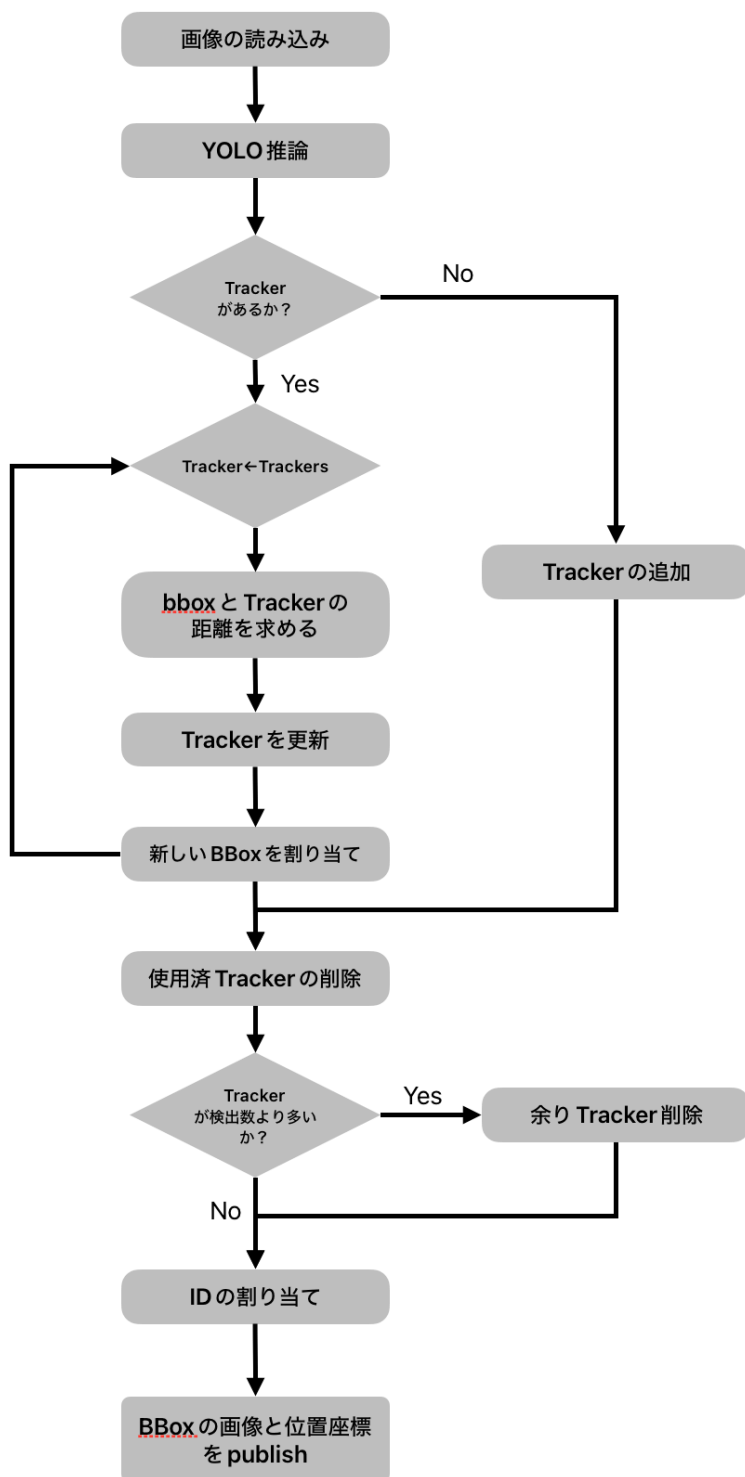


図 29 トラッキングアルゴリズム

4.2.5 果樹画像取得ソフトウェア

ROS と Python を用いて動画データから果樹ごとの画像を取得し、同時に位置座標の取得を行う果樹画像取得ソフトウェアを実装した。まず果樹検出モデルとトラッキングアルゴリズムを使用し、動画に対して物体検出を適用するノード (/tracker) を作成した (図 30)。このノードでは画像データ (/camera/color/image_raw) とその時の位置座標データ (/fix) を購読して果樹の検出およびトラッキングと位置座標の割り当てを行い、検出された単一果樹の画像、トラッキング ID、位置座標を含むデータ (/tracked_image_list) と、稼動しているトラッカーの ID (/tracker_heartbeats) を配信する。そして、トラッキングされた画像と追跡している果樹の ID を購読するノード (/tree_extraction) を作成した。このノードではトラッキングした果樹の画像を ID ごとに保存し、トラッキングしている ID が途切れた際に蓄積した画像のうち面積が最大のものを果樹画像として保存する。ただし、保存の際は設定した閾値を超える枚数を保存した場合のみ果樹画像として保存する。これにより、検出ミスによって数フレームのみ検出された画像を排除することが可能である。このノードで取得した果樹画像は /extracted_image というメッセージ名で配信する。そして、果実検出モデルなどを用いて果実などの計数を行うノード (/inference) で /extracted_image

を購読し、花芽検出 AI や果実検出 AI を適用することが可能である。

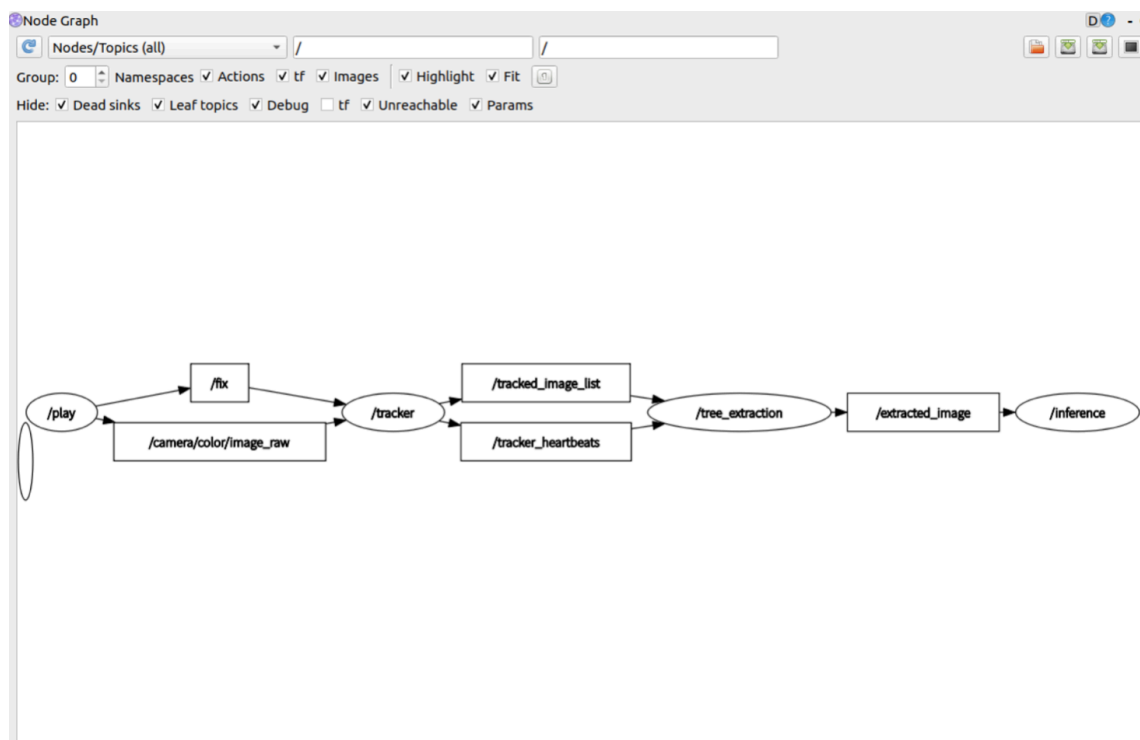


図 30 果樹画像取得ソフトウェア

4.2.6 データの取得

作成した果樹トラッキングシステムの開発の性能を検証するために、実際の圃場においてデータの取得を行った。本研究ではハイブッシュブルーベリー品種である「ブルークロップ」を対象とした。北海道余市町にあるブルーベリー農園であるアイケイファーム余市において、7月から8月にかけて作成したデータ取得システムを用いて2回走行し、動画データと位置データを取得した。7月のデータに映る果樹は果実が多く結実していたが、8月のデータに映る果樹はほとんど

の果実が収穫されていた。

4.2.7 評価方法

果樹トラッキングシステムの評価については果樹検出モデルによる検出精度とトラッキング精度について評価した。果樹検出モデルについては、モデル学習時の画像解像度と同じ解像度のテストデータに対して推論を行い、その結果について評価を行った。本研究では IoU が 0.45 を超えた場合のみ検出に成功したものとした。AI による物体検出の評価指標には平均適合率 (AP; Average Precision) および、F 値 (F1; F1-score)、適合率 (P; Precision)、再現率 (Re; Recall) に加えて IoU の平均値を導入した。IoU の平均値は、検出に成功した検出結果を用いて平均値を算出した。適合率、再現率、F 値、決定係数を評価する際の Confidence Score の閾値は、検証用データに対する推論結果の全クラスの F 値の平均値が最も高くなる値と、果樹トラッキングの際に使用した閾値である 0.6 の二つの場合に設定し、検証を行った。

また、果樹トラッキングの評価については目視によって動画から確認した果樹と、開発したシステムで取得した果樹画像を比較することで果樹の見落としや重複を判断し、果樹トラッキングの精度評価を行った。評価指標には適合率と再現率および、これらの調和平均である F 値を評価指標として検証した。検証に用いたデータは果樹検出 AI を作成した際の動画と、それとは別の日に撮影し

た動画の2つの動画を用いてトラッキングを行い、その精度検証を行った。

4.3 結果と考察

4.3.1 果樹検出 AI の評価

作成した果樹検出モデルによる推論結果の評価を行った（図 31、表 13、表 14）。640 ピクセルで学習されたモデルは 640 ピクセルで推論を行い、1280 ピクセルで学習されたモデルは 1280 ピクセルで推論を行った。検証には2種類のデータを使用し、果樹検出モデルの作成に用いたデータをテストデータ、別日に撮影したデータを未知データとして検証を行った。また Confidence Score の閾値は、検証用データに対する推論結果の全クラスの F 値の平均値が最も高くなる値と、果樹トラッキングの際に使用した閾値である 0.6 の二つの場合で検証を行った。表 14 において 0.6 に設定した結果については (0.6) と表記した。平均適合率の最大値はテストデータに対する 640 ピクセルの推論で 0.976 となり、未知データに対してはテストデータに対する 640 ピクセルの 0.919 となり、それぞれ高い値を示した。検出精度に関しては、テストデータに対する F 値の最大値は 0.955、未知データでは 0.905 となった。平均 IoU の最大値に関してはテストデータにおいては 0.855、未知データにおいては 0.810 となり高い値を示した。トラッキングに用いた Confidence score が 0.6 の場合は、再現率が低くなる代わりに適合率が高くなり、適合率の最大値は 0.988 となった。また、F 値を

比較すると 640 ピクセルの推論が 1280 ピクセルの推論の検出精度を上回る傾向にあった。



図 31 果樹検出 AI による推論結果の例

表 13 果樹検出 AI の評価（平均適合率）

	640・8月	1280・8月	640・7月	1280・7月
平均適合率	0.976	0.969	0.918	0.919

表 14 果樹検出 AI の評価（検出精度）

解像度・検証データ	P	R	F	平均 IoU
640・テスト	0.987	0.924	0.954	0.854
640・テスト (0.6)	0.988	0.924	0.955	0.855
1280・テスト	0.964	0.942	0.882	0.830
1280・テスト (0.6)	0.973	0.836	0.899	0.829
640・未知	0.945	0.869	0.905	0.802
640・未知・0.6	0.962	0.768	0.854	0.810
1280・未知	0.943	0.828	0.882	0.804
1280・未知・0.6	0.958	0.687	0.800	0.799

精度評価の結果から、果樹検出 AI を用いて複数の果樹が映る画像から果樹を高精度に検出することを可能とした。平均 IoU についても高い値を示しており、検出した果樹画像を用いることで果樹ごとの花芽や果実の計数を実現できることを示唆している。また、640 ピクセルの画像推論が 1280 ピクセルの検出結果を上回った要因として、学習時に解像度を落とした画像を用いたことに起因すると考えられる。テストデータと比べて、未知データに対する検出精度は低い傾

向にあった。今後天候や場所の異なる複数のデータを用意しモデルの学習を行うことで、様々な環境下での果樹の検出が実現すると考えられる。

また、Confidence score を 0.6 に設定した場合に適合率が高い値を示し、最大で 0.988 を示した。本研究ではカルマンフィルタを用いたトラッキングアルゴリズム適用し果樹を追跡する。そのため、果樹の見落としよりも誤検出を減らすことが求められるため、適合率の高いトラッキングに適したモデルを作成できたと言える。

4.3.2 果樹トラッキングの評価

本研究で作成した果樹トラッキングシステムを用いて果樹ごとの画像の取得を行い、その精度評価を行った（図 32、表 15）。動画解像度を 640 ピクセルと 1280 ピクセルの 2 種類に設定し、果樹検出モデルの学習・推論解像度は動画解像度に一致するようにし、精度検証を行った。精度検証には 2 種類のデータを使用し、果樹検出モデルの作成に用いたデータをテストデータ、別日に撮影したデータを未知データとして検証を行った。最も精度の高い結果はテストデータに対して 640 ピクセルで推論した場合であり、見落としした果樹が 1 本であり重複した果樹が 1 本であった。また、未知データに対する結果においては 640 ピクセルで推論した結果で見落としした果樹が 2 本、重複が 5 本となった。未知データに対する結果においても再現率が 0.979 の値を示し、見落としの少ない検出

を実現した。

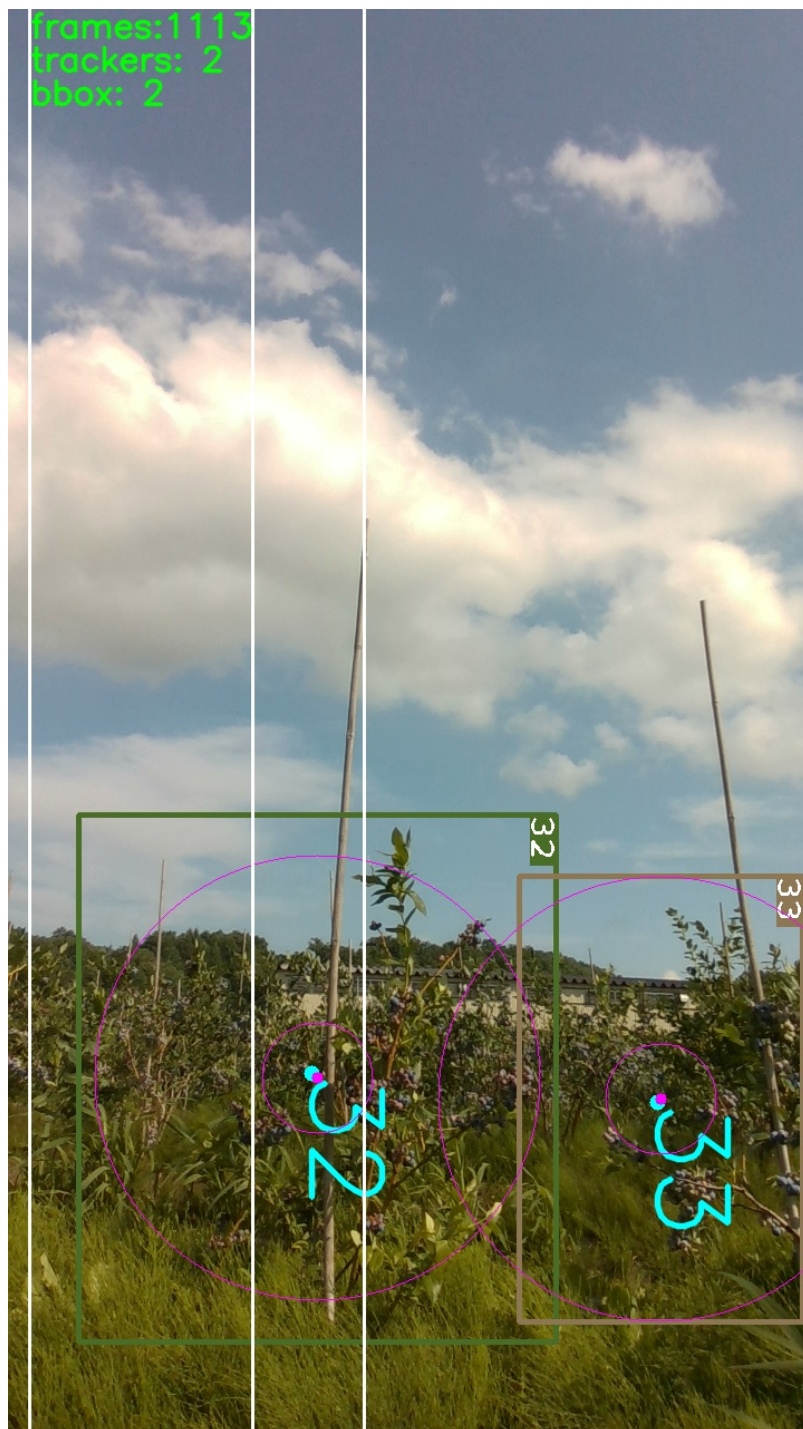


図 32 果樹トラッキングの例

表 15 果樹トラッキングの結果

解像度・検証データ	検出数	見落とし	重複	P	Re	F
640・テスト	98	1	1	0.989	0.989	0.989
1280・テスト	95	3	0	1.000	0.969	0.984
640・未知	102	7	3	0.950	0.979	0.965
1280・未知	93	2	5	0.967	0.928	0.947

以上の結果から、果樹画像を取得するのに十分な精度が得られたこと示唆された。図 32 の青点はカルマンフィルタによる予測値である。トラッキング結果の動画を確認すると果樹検出モデルの誤検出がない場合は、果樹のトラッキングが正確に行われていた。一方で果樹の誤検出が発生した場合には、見落としや重複が発生した。例えば 640 ピクセル・テストデータに対する結果が最も精度が高くなったが、この場合においても重複および見落としが発生した。重複した果樹を図 33 に示す。これは果樹検出モデルによる誤検出により 2 本の果樹の間を検出した場合であり、見落としとともに重複が発生した。また、1280 ピクセル・未知データに対する結果で多く発生した果樹の重複は、果樹検出モデルが検出に連続して失敗したことでカルマンフィルタによるトラッキングが途中で終了してしまい、果樹の ID が変わってしまった場合に発生していた。これらの問題は果樹検出モデルの検出精度を向上させたモデル作成し、誤検出や検出漏れ

を改善することで解決すると考えられる。今後、データセットの拡張や別の物体検出 AI を利用して検出精度の高い果樹検出モデルを作成することで、全ての果樹の画像を正確に取得できると考えられる。

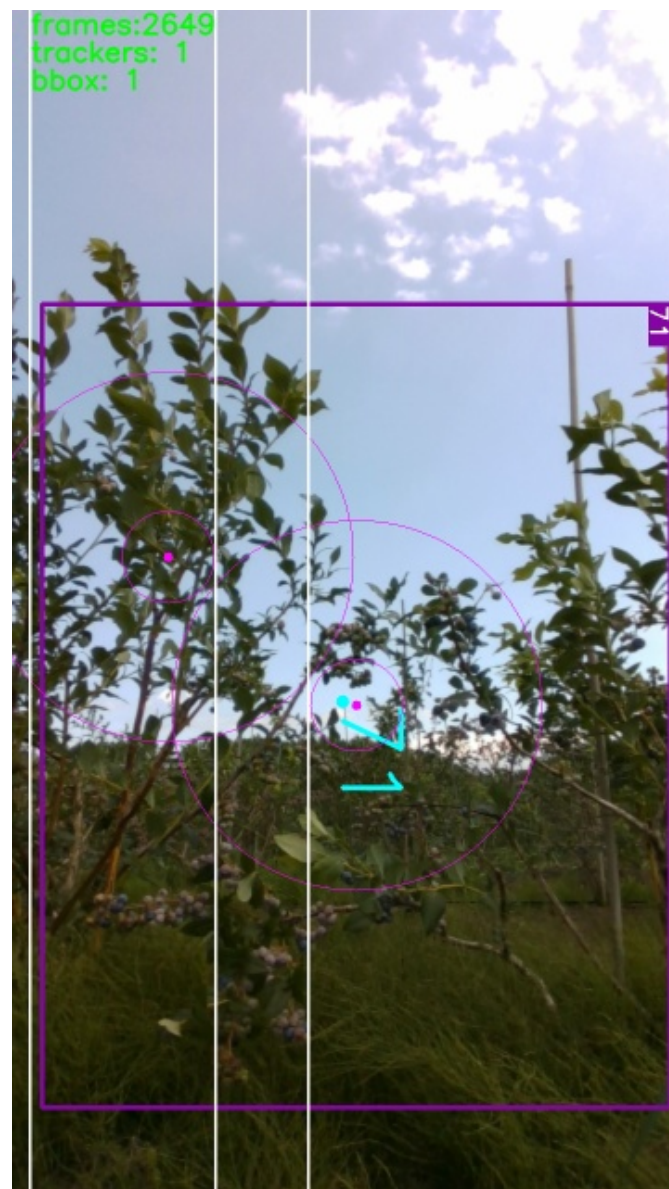
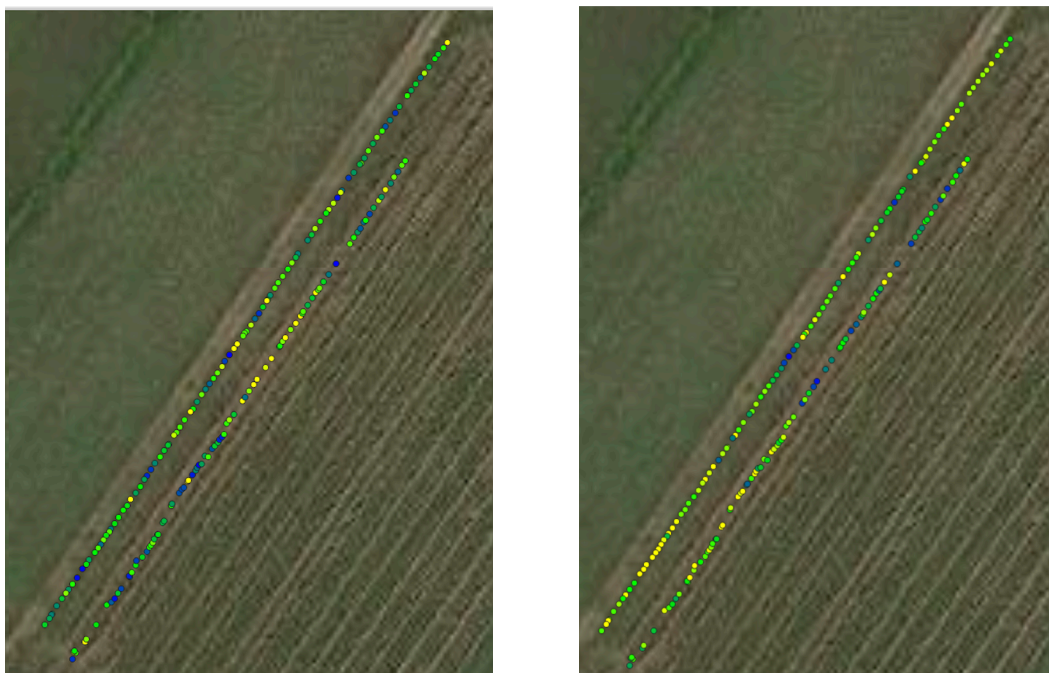


図 33 誤検出の例

4.3.3 果実数と成熟果実率のマップ化

果樹トラッキングシステムによって取得した果樹ごとの画像に対して果樹検出モデルを適用し、果実数と成熟果実率のマッピングを行った(図 34)。果樹検出モデルには第 2 章で作成した YOLOv5s をベースに 1280 ピクセルで学習したモデルを使用した。



(a) 果実数

(b) 成熟果実率

図 34 果実数および成熟果実率のマップ化

マップから果実数や成熟果実率の分布にばらつきが生じているのを確認することができた。このマップによって適切な収穫順序や収穫時期の決定が実現すると期待される。同様に、花芽検出モデルを用いて花芽数のデータを取得するこ

とで、今後結実率の可視化への利用が期待される。一方で、使用した果樹検出モデルは別のカメラで撮影された画像を用いて作成したものであり、検出結果は正確ではない。今後、動画から作成したデータセットを用いて花芽検出モデルや果樹検出モデルを作成することで、正確に圃場データのマップ化が実現すると期待される。

4.4 結論

この章では物体検出を用いて花芽や果実を自動計数するために、果樹ごとの画像を取得するためのシステム開発を行った。まず、圃場内の動画データおよび位置座標を取得するデータ取得システムを開発した。そして、物体検出 AI を用いた果樹検出モデルの作成と、動画に対して果樹検出モデルを適用するためのソフトウェアを開発した。これらを用いて実際の圃場でデータの取得と、果樹の検出および果樹トラッキングの精度検証を行った。果樹検出モデルの平均適合率と平均 IoU の最大値はそれぞれ 0.976、0.855 となり高精度な果樹の検出を実現した。また果樹トラッキングの結果は最も良かった場合で F 値が 0.989 となり、果樹ごとの画像を高精度に取得することを可能とした。また、果実数及び成熟果実率のマップ化の一例を示し、本システムの応用可能性を示した。本研究で作成した果樹トラッキングシステムを用いて、今後花芽や果実の自動計数が実現すると期待される。

第 5 章 総括

ブルーベリー栽培において花芽の形成や、果実の結実および熟成について圃場の細かなモニタリングと管理が重要である。花芽や果実のモニタリングは花芽数による早期収量予測、成熟果実数による収穫時期や収穫順序の決定など幅広い営農管理への利用が期待される。しかし、果樹 1 本あたり数百から数千もの花芽や果実をつけるため、現状の人手によるモニタリングではデータの取得には限界がある。そのため圃場モニタリングの自動化が求められており、これには物体検出 AI が有用であると考えられる。そこで本研究ではブルーベリー圃場のモニタリングの自動化を目的として、物体検出 AI を用いた果樹ごとの花芽と果実の検出と分類、画像を用いた果実の推定可能性の検討および果樹ごとの画像取得のためのシステム開発を行った。

第 2 章では物体検出 AI を用いたブルーベリーの花芽と果実の計数および分類の有用性を評価した。まず、YOLOv5 と YOLOv8 を使用し、ブルーベリーの花芽数と果実数を推定した結果、凍害を受けた花芽に関しては改善の余地があるものの、一つの果樹が収まる画像から花芽と果実の検出・分類が可能であることが示された。また、640 ピクセルの低解像度画像を用いた推論においても高い決定係数を示した。単回帰式を用いた画像中の花芽と果実の計数においては、過学習の問題が生じたものの MAPE の最小値は花芽で 10.79%、果実で 23.84% とな

った。一方で、物体検出 AI の適用には画像中の果樹を手動でトリミングする必要があったため、今後の課題として例えば物体検出 AI を用いて個々の果樹を自動でトリミングするなどの手法開発が求められる。この手法については第四章で検討を行った。

第 3 章では結実期のブルーベリー果樹を撮影した画像では葉に隠れてしまう果実があることを考慮し、画像中の果実数から実際の果実数が推定可能かどうかについて検証した。その結果、画像に映る果実数と実際の果実数には正の相関 ($R=0.96$, $p<0.001$) があることを明らかにした。また単回帰式との回帰誤差は MAPE で 21.2% となり、画像から果実数を推定することの妥当性を示した。さらに、高精度かつ効率的な圃場モニタリングの自動化を実現するために、画像から果実数を推定する際に葉の被覆率、撮影方角、樹列の両側からの撮影が推定精度に及ぼす影響を解析した。その結果、葉の被覆率が推定精度に大きく影響することが明らかとなった。このことから今後、画像を用いて高精度に果実数を推定するためには、セマンティックセグメンテーション AI などによって算出された果樹の葉の被覆率の利用が有効であることが考えられる。一方で、撮影方角の考慮や樹列の両側から撮影することは推定精度に影響がなく、画像を用いた果実計数の際は同一の果樹を樹列の右側と左側に関わらず、一度撮影するだけで十分である可能性が示された。これにより、撮影のための作業経路を短縮でき、圃場

全体の効率的なモニタリングが実現可能であると思われる。

第 4 章では物体検出を用いて花芽や果実を自動計数するために、果樹ごとの画像を取得するためのシステム開発を行った。まず、圃場内の動画データおよび位置座標を取得するデータ取得システムを開発した。そして、物体検出 AI を用いた果樹検出モデルの作成と、動画に対して果樹検出モデルを適用するためのソフトウェアを開発した。これらを用いて実際の圃場でデータの取得と、果樹の検出および果樹トラッキングの精度検証を行った。果樹検出モデルの平均適合率と平均 IoU の最大値はそれぞれ 0.976、0.855 となり高精度な果樹の検出を実現した。また果樹トラッキングの結果は最も良かった場合で F 値が 0.989 となり、果樹ごとの画像を高精度に取得することを可能とした。また、果実数及び成熟果実率のマップ化の一例を示し、本システムの応用可能性を示した。本研究で作成した果樹トラッキングシステムを用いて、今後花芽や果実の自動計数が実現すると期待される。

以上の研究内容から、物体検出 AI を用いた花芽数と果実数の推定が実現可能であることを示した。今後本研究の内容をさらに発展させることで、ブルーベリー圃場の自動モニタリング実現し、花芽の計数による早期収量予測、受粉度合いの把握、収穫適期や収穫順序の判断など様々な現場への応用が期待される。

謝辞

本研究は 2021 年 4 月から 2023 年 12 月まで、博士課程の学位請求論文として、北海道大学大学院農学院生産フロンティアコース生物生産工学ユニットビークルロボティクスにおいて博士課程在学中に行われたものです。研究着手当初より、ご指導、ご鞭撻を賜ったビークルロボティクス研究室の岡本博史准教授に厚く御礼申し上げます。また、研究に多大なご助言、ご協力をいただいたビークルロボティクス研究室の招聘教員の松浦健太郎さん、学部 4 年の高杉翔君、北海道大学大学院獣医学研究院・感染症学教室の中村隼人氏に感謝申し上げます。また、アイケイファーム余市株式会社様には、実験環境の提供に加え、多大なご助言、ご協力を頂きました。厚く御礼申し上げます。本研究は、JST 次世代研究者挑戦的研究プログラム JPMJSP2119 の支援を受けたものです。

参考文献

Arrington, M. and L.W. DeVetter (2018) Increasing honey bee hive densities promotes pollination and yield components of highbush blueberry in western washington, HORTSCIENCE, 53(2):191-194.

Gonzalez, S., C. Arellano and J.E. Tapia (2019) Deepblueberry: quantification of blueberries in the wild using instance segmentation, IEEE Access, 7:105776-105788.

Google (2023a) Google cloud documentation, <<https://cloud.google.com/docs/>>, browsed on Sept. 28, 2023.

Google (2023b) Google maps platform documentation, <<https://developers.google.com/maps/documentation/>>, browsed on Sept. 28, 2023.

He, K., G. Gkioxari, P. Dollar and R. Girshick (2018) Mask R-CNN, arXiv:1703.06870v3 [cs.CV] 24 Jan 2018.

HumanSignal (2023) labelImg, <<https://github.com/HumanSignal/labelImg/>>, browsed on May 1, 2023.

板倉健太, 林 拓哉, 野秋 収平, 上脇 優人, 細井 文樹 (2022) 深層学習を用いた根菜類の個数カウンティングによる収量推定法の開発, AI・データサイエ

- Li, G., R. Suo, G. Zhao, C. Gao, L. Fu, F. Shi, J. Dhupia, R. Li and Y. Cui (2022)
Real-time detection of kiwifruit flower and bud simultaneously in orchard using
YOLOv4 for robotic pollination. Computers and Electronics in Agriculture,
193:106641.
- MacEachern, C.B., T.J. Esau, A.W. Schumann, P.J. Hennessy and Q.U. Zaman
(2022) Detection of fruit maturity stage and yield estimation in wild blueberry
using deep learning convolutional neural networks, Smart Agricultural
Technology, 3:100099.
- Mirhajia, H., M. Soleymania, A. Asakereh and S.A. Mehdizadeh (2021) Fruit
detection and load estimation of an orange orchard using the YOLO models
through simple approaches in different imaging and illumination
conditions, Computers and Electronics in Agriculture, 191(2021):106533.
- Redmon, J., S. Divvala, R. Girshick and A. Farhadi (2016) You only look once:
unified, real-time object detection, IEEE Conference on Computer Vision and
Pattern Recognition (CVPR), 2016:779-788.
- Ronneberger, O., P. Fischer and T. Brox (2015) U-Net: convolutional networks
for biomedical image segmentation, arXiv:1505.04597v1 [cs.CV] 18 May 2015.

ROS.org (2023b) Robot operating system, <<https://wiki.ros.org/ja>>, browsed on Oct. 1, 2023.

Salvo, S., C. Muñoz, J. Ávila, J. Bustos, M. Ramírez-Valdivia, C. Silva and G. Vivallo (2012) An estimate of potential blueberry yield using regression models that relate the number of fruits to the number of flower buds and to climatic variables, *Scientia Horticulturae*, 133:56-63.

Tian, Y., G. Yang, Z. Wang, H. Wang, E. Li and Z. Liang (2019) Apple detection during different growth stages in orchards using the improved YOLO-V3 model, *Computers and Electronics in Agriculture*, 157:417-426.

Towne, W.F. and W.H. Kirchner (1989) Hearing in honey bees: detection of air-particle oscillations, *Science*, 244(4905):686-688.

Ultralytics (2023a) Ultralytics, <<https://github.com/ultralytics/ultralytics/>>, browsed on May 1, 2023.

Ultralytics (2023b) YOLOv5, <<https://github.com/ultralytics/yolov5/>>, browsed on May 1, 2023.