

令和7年度
江別市大学連携調査研究事業

画像処理技術を用いた
ハウス農業スマート化の実現可能性調査
実施報告書

2026年3月22日

北海道情報大学
湯村 翼, 栗原 純一

目次

1. 調査研究概要	3
2. ARマーカ―を用いたビニールハウスの窓開閉監視システムの開発と調査	4
2.1 はじめに	4
2.2 システム構成	4
2.2.1 システム概要	4
2.2.2 ARマーカ―による開閉検出	5
2.2.3 実行プログラム	5
2.2.4 台座の設置	6
2.3 ARマーカ―に関する検証	7
2.3.1 ArUcoマーカ―とAprilTag	7
2.3.2 検出距離の検証	8
2.3.2.1 検証方法	8
2.3.2.2 検証結果	9
2.3.3 角度検出の検証	10
2.3.3.1 検証方法	10
2.3.3.2 検証結果	10
2.3.4 暗所での検出の検証	11
2.3.4.1 検証方法	11
2.3.4.2 検証結果	11
2.4 考察	12
3. トマトの成熟度モニタリングシステムの開発と調査	13
3.1 はじめに	13
3.2 開発プロセス	13
3.2.1 データ準備と前処理	13
3.2.2 モデル設計と学習アプローチ	13
3.2.3 モデル性能評価	13
3.3 実装	14
3.3.1 学習・変換プログラム	14
3.3.2 リアルタイム推論プログラム	14
3.4 実験	15
3.4.1 実験方法	15
3.4.2 実験結果	16
4. おわりに	18
4.1 ARマーカ―を用いたビニールハウスの窓開閉監視システムにおける課題	18
4.2 トマトの成熟度モニタリングシステムにおける課題	18
4.3 まとめ	18

1. 調査研究概要

本報告書は、画像処理技術を用いてハウス農業のスマート化を実現するための二つの主要な研究課題について、その実現可能性を調査し、開発と検証結果をまとめたものである。

2節「ARマーカーを用いたビニールハウスの窓開閉監視システムの開発と調査」では、生産者の負担軽減と高温障害の早期把握を目的とし、換気窓の開閉状態とハウス内温度を遠隔から監視可能なシステムを構築した。換気窓に貼付されたARマーカーの検出数に基づき開閉状態を判定し、温度情報と共にメッセージアプリ(Slack)へ通知する仕組みであり、検出距離や角度に関する検証により、システムが実用的な環境で機能する有用性を確認した。

3節「トマトの成熟度モニタリングシステムの開発と調査」では、収穫作業の効率化を目指し、画像認識技術を用いてトマトの成熟度(完熟, 半熟, 未熟)を自動で推定するプログラムを開発した。深層学習モデル(MobileNetV2ベースの転移学習)を構築し、Raspberry Pi上でリアルタイム推論アプリケーションとして実装した。モデルは平均正解率約86.92%を達成し、完熟トマトの選別に実用可能性があることを確認した。

2.ARマーカ―を用いたビニールハウスの窓開閉監視システムの開発と調査

2.1 はじめに

ビニールハウスにおける施設栽培では、ハウス内の温度管理が作物の生育に大きな影響を与える。ハウス内は温室効果により外気より温度が高くなりやすく、高温状態が続くと作物が弱るなどの高温障害が発生する可能性がある。そのため、換気窓を開閉することで温度を適切に調整する必要がある。

しかし、天候の変化によりハウス内温度は大きく変動するため、生産者は一日に何度も現地で温度や換気窓の状態を確認しなければならず、その作業は大きな負担となっている。近年では換気を自動で行う自動開閉装置が普及しているが、故障が発生した場合には生産者が異常に気づくことが遅れ、ハウス内の高温化によって作物に被害が及ぶおそれがある。

そこで本研究では、自動開閉装置の異常を早期に把握できるよう、換気窓の開閉状態とハウス内温度を遠隔から監視可能な装置の開発を目的とする。

2.2 システム構成

2.2.1 システム概要

本研究で開発したシステムは、ハウスの側面に、フィルム巻き上げ式の換気窓が設置されていることを前提に、開閉状態およびハウス内温度を遠隔から監視することを目的としている。システムの構成を図1に示す。

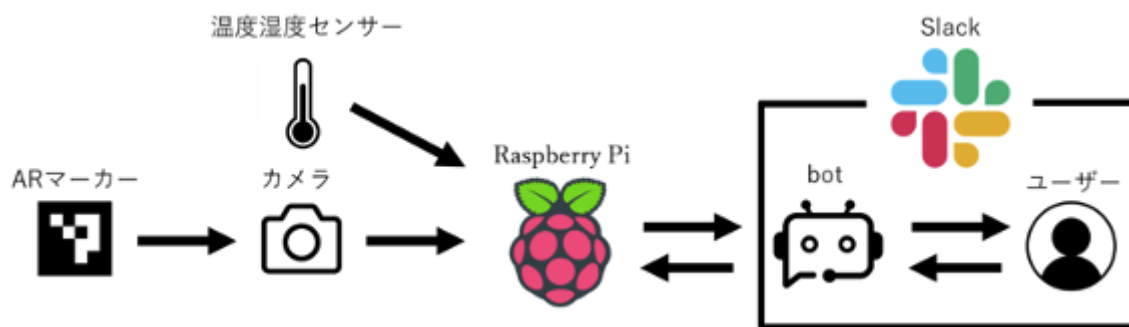


図1 システム構成図

本システムでは、換気窓に複数のARマーカ―を縦一列に貼付し、カメラによってこれらのマーカ―を撮影する。換気窓の開閉に伴ってカメラに映るマーカ―の数が増減することを利用し、検出されたマーカ―の数から窓の開閉状態を判定する。

また、ハウス内には温度センサを設置し、リアルタイムで温度を計測する。取得した開閉状態および温度情報は小型コンピュータであるRaspberry Piで処理され、通信機能を用いて遠隔地の利用者へ送信される。これにより、利用者は現地に赴くことなく、ハウス内の状態を確認することが可能となる。

カメラにはRaspberry Pi Camera Module 3を使用した。本モジュールは1200万画素のSony IMX708センサーを搭載し、オートフォーカス及びHDR撮影に対応している。

温度センサーにはDHT22を使用した。本センサーは温度および湿度を同時に測定可能なデジタル出力型センサーであり、 $\pm 0.5^{\circ}\text{C}$ の誤差で温度を測定できる。

2.2.2 ARマーカによる開閉検出

本システムでは、換気窓の開閉状態を検出するためにARマーカを用いた。ARマーカとは、白黒の二値パターンで構成された正方形の画像マーカであり、カメラ画像から容易に識別できる特徴を持つ。主に拡張現実(AR)において物体の位置や姿勢を推定するために用いられる。

換気窓には複数のARマーカを縦一列に配置した。窓の開閉に伴ってカメラに映るマーカの数が増えるため、検出されたマーカの個数を基に開閉状態を判定することができる。窓が完全に閉じている場合は多くのマーカが認識される一方、窓が開くにつれて認識されるマーカ数は減少する。この原理を図2に示す。

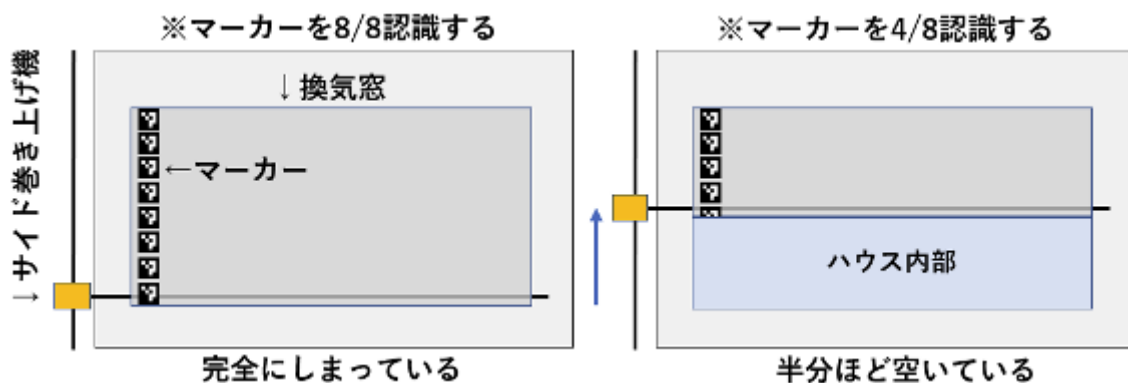


図2 換気窓の開閉状態とマーカ認識数の関係

2.2.3 実行プログラム

本研究ではプログラミング言語としてPythonを使用した。Raspberry Piは一定時間おきにカメラからの画像と温度の情報を取得する。カメラ画像からマーカを検出し、マーカのIDと検出個数、取得した画像をメッセージアプリであるSlackに通知する。

Slackとは主にビジネス向けに使用されているチャットツールである。本システムでは異なるソフトウェアやWebサービスどうしをインターネットを介して接続するインターフェースであるAPIを使用し、Raspberry PiからSlackへの接続を実現している。Slackの通知画面を図3に示す。

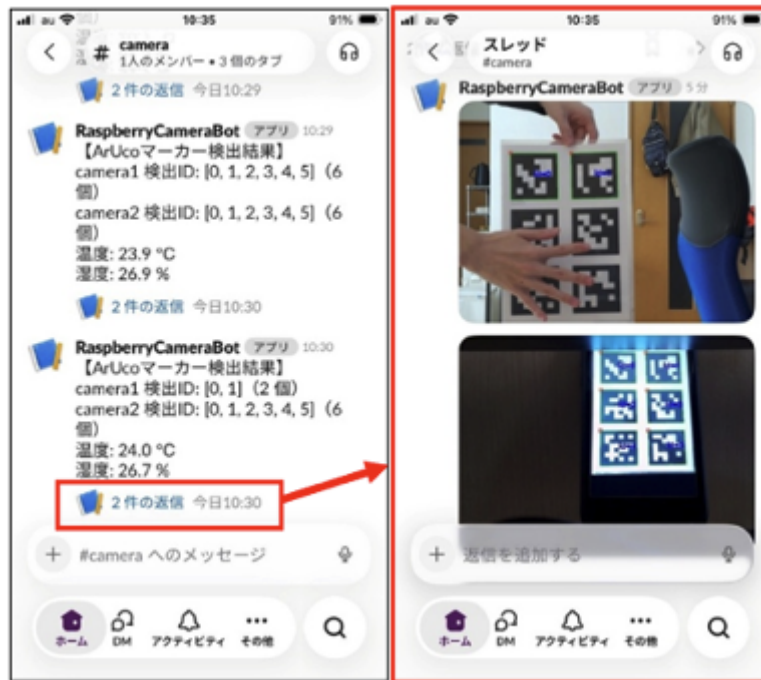


図3 Slackの通知画面

2.2.4 台座の設置

木材の板を加工し、台座を作成した。台座の上部には箱を設け、その中にRaspberry Piとカメラを収納している。温度センサーは箱に設けられている穴に通し、箱の外側に設置している。またカメラの向いている面にアクリルボードを使用することでカメラの視界を確保している。台座の図を図4に示す。

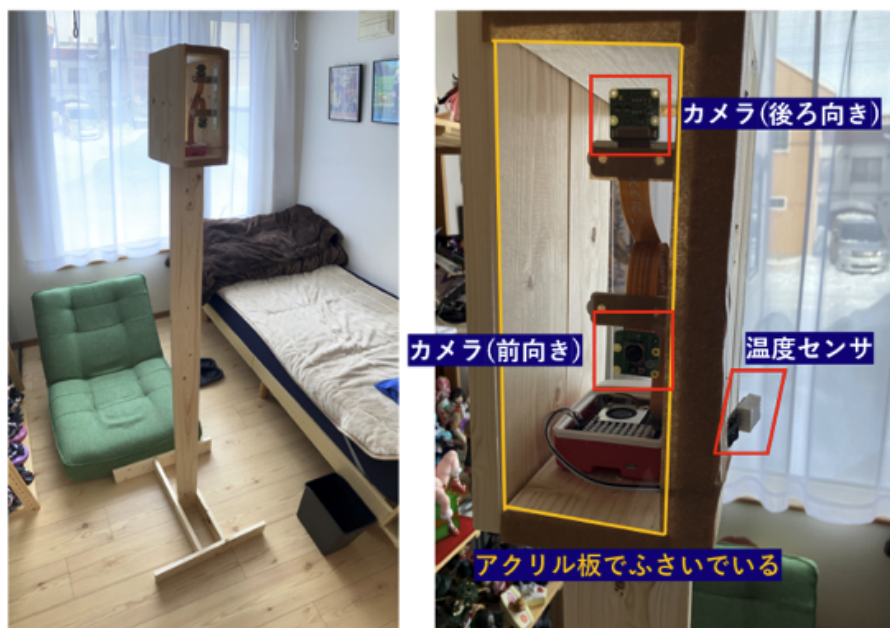


図4 台座の図

台座は、ハウス内の正面から見て中央付近に設置する想定で作成した。またフィルム巻き上げ式の換気窓は、左右の側面に存在していることを想定しているため、2台のカメラを使用することによって左右両方の換気窓を撮影することができる。設置想定図を図5に示す。

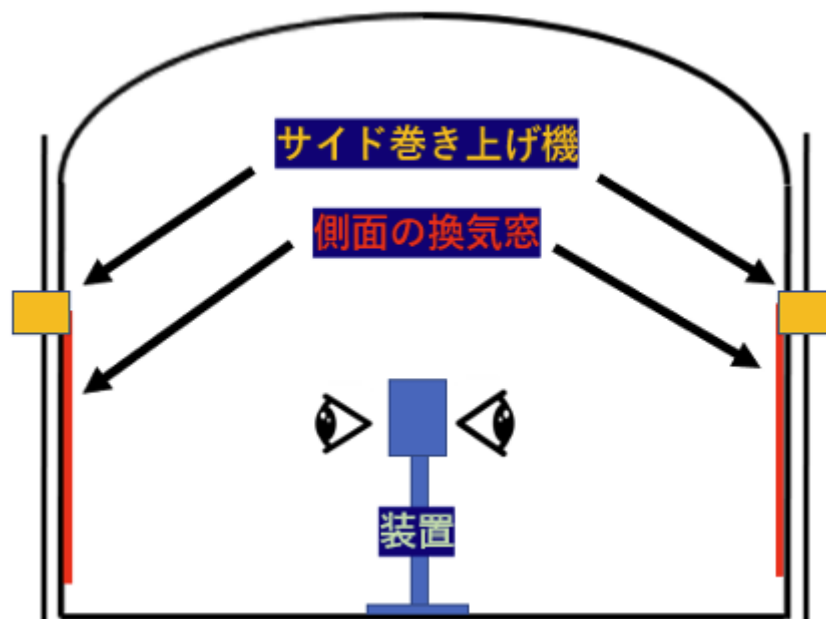


図5 設置想定図

2.3 ARマーカーに関する検証

2.3.1 ArUcoマーカーとAprilTag

ARマーカーは様々なものが開発されている。本研究では、ArUcoマーカー、AprilTagの2種類のARマーカーを使用した。

ArUcoマーカーは2011年にスペインのコルドバ大学の研究チームが開発したARマーカーである。このマーカーは計算量が少ないためCPUへの処理負荷が小さく、それでいてカメラのフレームレートに合わせ、リアルタイムで計算処理ができるという特徴がある。

AprilTagは2011年にアメリカのミシガン大学の研究チームが開発したARマーカーである。このマーカーはArUcoマーカーと比較し、検出アルゴリズムを細かく設定することができる点や、ハミング距離が大きく設定されており、誤検出に強いという特徴がある。ハミング距離とは、2つのビット列(0と1の並び)が、何ビット違っているかを表す数のことである。この値が高いほど、多少の悪条件や誤差があっても正しく動作し続ける強さ(ロバスト性)を持っていると判断できる。

この2種類のARマーカーは、マーカーの形が非常に似ており、かつ産業用ロボットの誘導やドローンの位置推定など様々な場面で広く使われているため本研究で使用した。

今回作成した台座は、ハウス内の正面から見て中央付近に設置する想定であると述べたが、各企業が販売しているビニールハウスの図面を確認した結果、中央から側面まで、3～5mの距離があることが分かった。また、ハウス内の作物の位置関係の問題から、マーカーを斜めに撮影しなければならない可能性や、屋外であるため、日射量の関係でマーカーを認識できない可能性が想定される。そのため本研究では、マーカーの検出距離、検出角度、蓄光テープを用いたマーカーの作成に関する検証を行った。

2.3.2 検出距離の検証

2.3.2.1 検証方法

屋外環境において ArUco マーカーおよび AprilTag を対象とした検出距離の検証を行った。検出距離検証の様子を図6に示す。



図6 検出距離検証の様子

カメラを固定した状態で各マーカーをカメラ正面に配置し、カメラからの距離を 2 m から 1 m 間隔で徐々に離して撮影を行った。各距離について 5 回の撮影を実施し、その距離においてマーカーが検出された回数を記録した。得られた結果を基に、検出率を「総検出個数／5」として算出した。

検証ではArUcoマーカーとしてArUco4x4とArUco6x6の2種類を、AprilTagとしてTag16h5とTag36h11の2種類の計4種類のマーカーを使用した。Arucoマーカーにおける 4×4 および 6×6 とは、内部のビットグリッドの大きさを示しており、4×4 は 16 ビット、6×6 は 36 ビットの情報を保持することを意味する。AprilTagにおけるTag16h5 および Tag36h11 という名称は、ビットグリッドの大きさと最小ハミング距離を表している。例えばtag16h5 は 16 ビットの情報(4×4のグリッド)を持ち、ハミング距離が少なくとも 5 以上になるよう設計されたタグであることを意味する。一般的にマーカーのグリッドが少なければ、マス目が大きいいため遠い距離からでも検出しやすくなり、グリッドが多ければハミング距離が大きくなり他のマーカーと間違える誤検出が少なくなるという特徴がある。そのため同種のマーカーからも情報量の違う2種類のマーカーを用意し、検出可能な距離に違いがあるか検証した。

2.3.2.2 検証結果

検証結果をグラフ化したものを図7に示す。

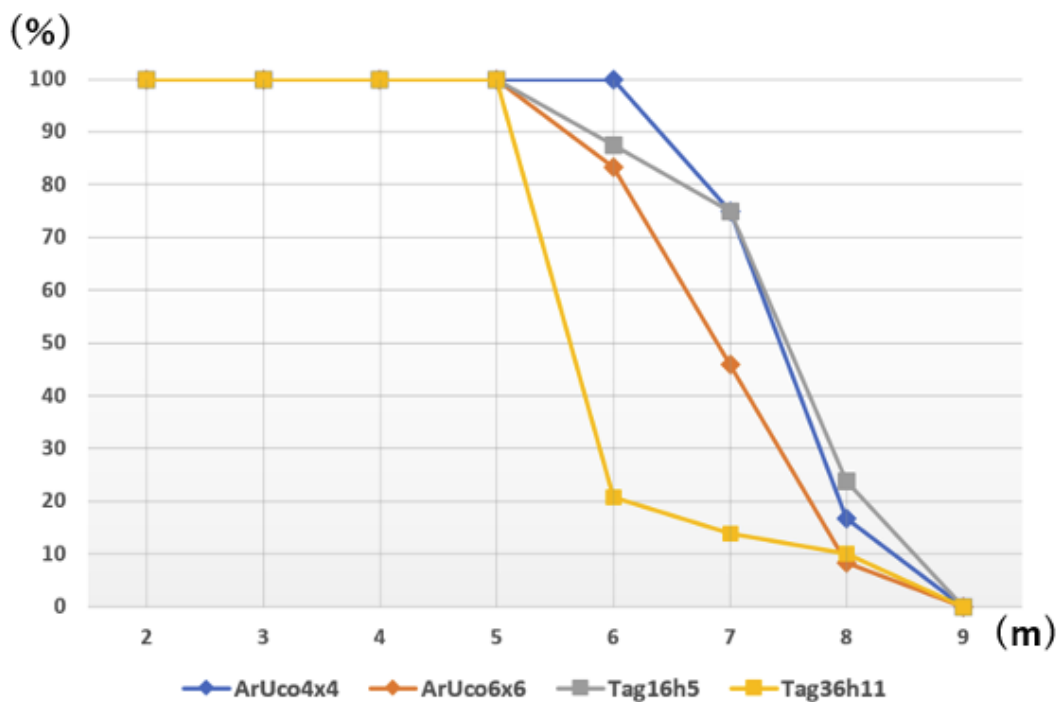


図7 検出距離検証の結果

検証の結果, すべてのマーカーにおいて5 mまでは検出率が100 %であった. また6 mでは, ArUco4x4のみが検出率100 %を維持し, それ以外のマーカーは検出率が低下した. 距離の増加に伴い検出率が徐々に低下し, 9 mではすべてのマーカーが検出不能となった. そして, Tag16h5に関しては, 用意したマーカーではない屋外の風景をマーカーとして誤検出する現象が常が発生した. 他のマーカー検出では誤検出は一度も発生しなかった. Tag16h5の誤認識の様子を図8に示す.



図8 Tag16h5の誤検出の様子

2.3.3 角度検出の検証

2.3.3.1 検証方法

検出角度に関する実験ではカメラから1mの距離にマーカを設置し、分度器を用いてマーカを検出角度を計測した。マーカをカメラの正面から徐々に傾け、すべてのマーカが検出可能な角度を計測した。実験環境を図9に示す。

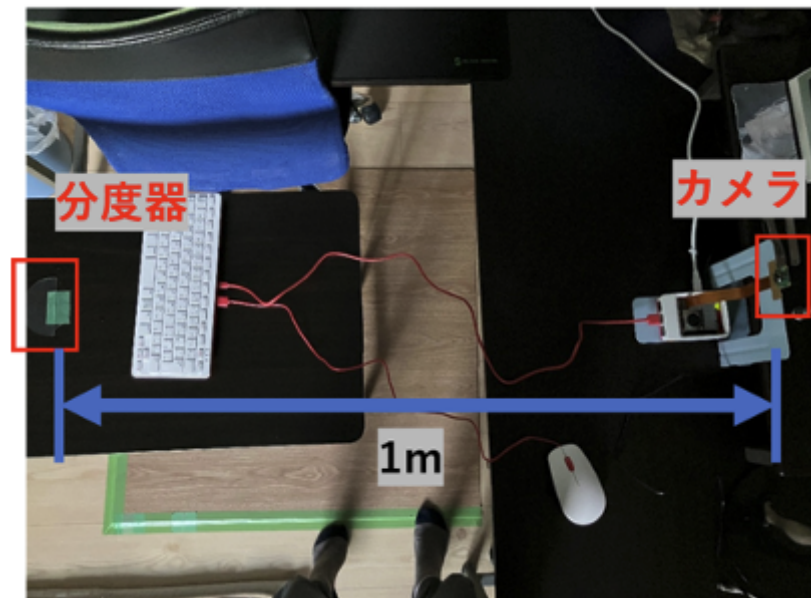


図9 検出角度検証の環境

2.3.3.2 検証結果

検証の結果、ArUcoマーカとAprilTagのいずれも、カメラ正面に対して約70度の角度まで検出が可能であった。それ以上の角度では、直角に近くなるにつれて検出が不安定となった。検出の様子を図10に示す。

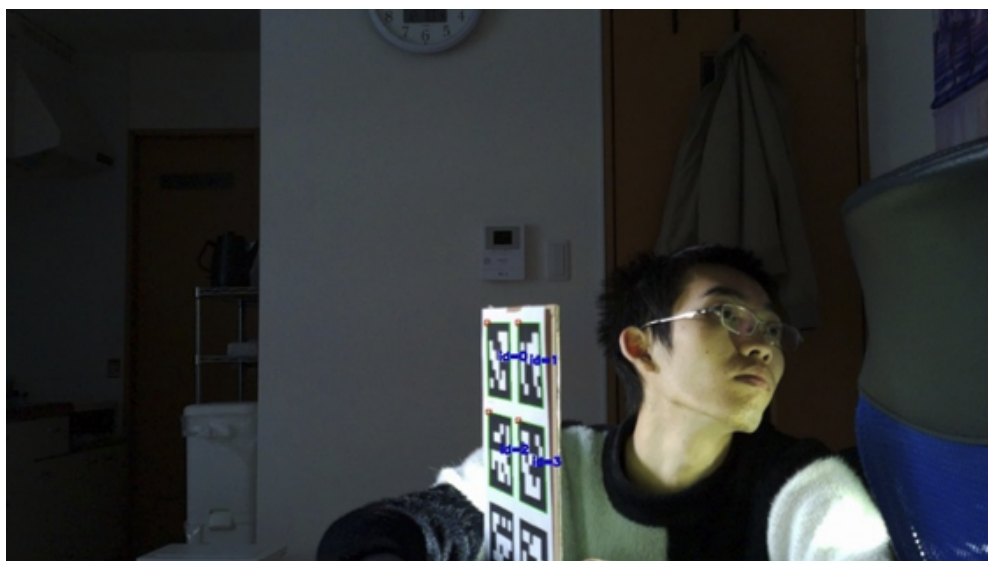


図10 検出の様子

2.3.4 暗所での検出の検証

2.3.4.1 検証方法

屋外は天候の影響により日中でも暗くなる場合がある。光量が少ない場合カメラのピントがマーカに合わず、またマーカの白と黒の判定があいまいになり、検出不能になるケースが確認された。そこで、蓄光テープをマーカの枠と、白いグリッドに張ることで、光量の少ない環境下でもマーカの検出が可能になるか検証した。蓄光マーカに2000ルーメンのライト光を一分間当て続けたのち、照明を消した状態で、蓄光マーカをカメラで撮影した。作成した蓄光マーカを図11に示す。

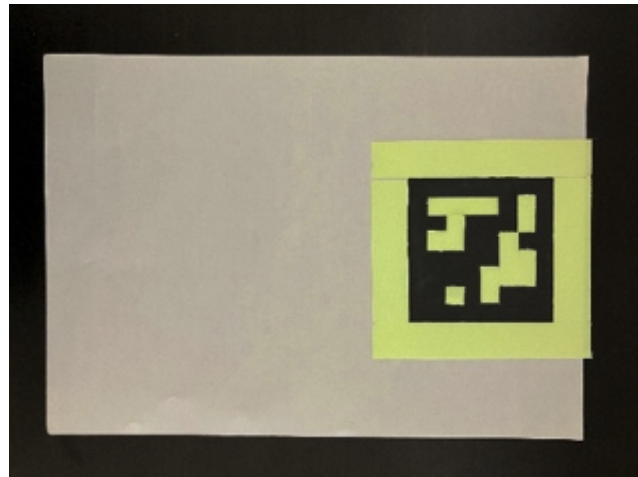


図11 蓄光マーカ

2.3.4.2 検証結果

ArUco マーカおよび AprilTag のいずれも検出が可能であることを確認した。完全な暗所での検証では蓄光マーカの蓄光量の減少により、およそ2時間後に検出不能となった。蓄光マーカを検出した様子を図12に示す。

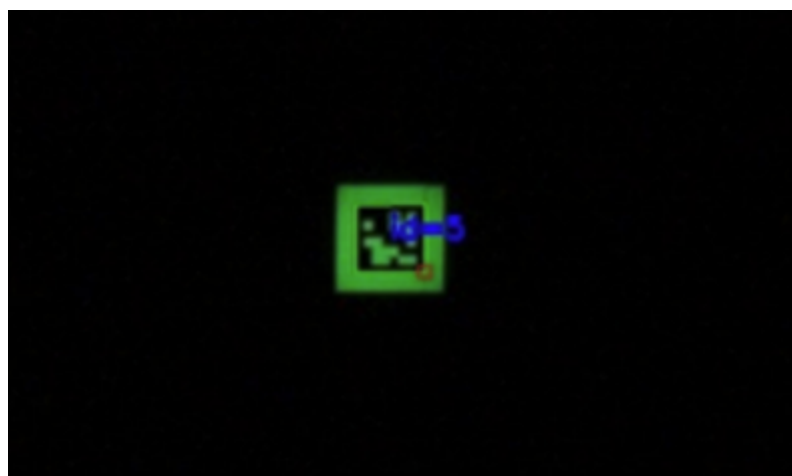


図12 暗所での蓄光マーカの検出の様子

2.4 考察

検出距離検証の結果は、5mまでの距離はすべてのマーカ―が検出可能であった。この結果から当初想定していた、ハウス内の中央付近に台座を設置し、3～5mの距離がある側面の換気窓に貼られているマーカ―をカメラで検出することが可能であるといえる。

検出距離実験で使用したTag16h5の誤検出について、原因は、彩度がはっきりしない白と黒の色の並びをマーカ―として認識したためと考察した。AprilTagはArUcoマーカ―と比較し、ハミング距離や彩度の許容値といった検出アルゴリズムを細かく設定することができる。検出距離検証の際にはそうした設定を行わなかった。後日自室で、ハミング距離を下げ、彩度がより大きい値のみ検出する設定を追加し検証したところ、誤検出をなくすことに成功した。検出の様子を図13に示す。

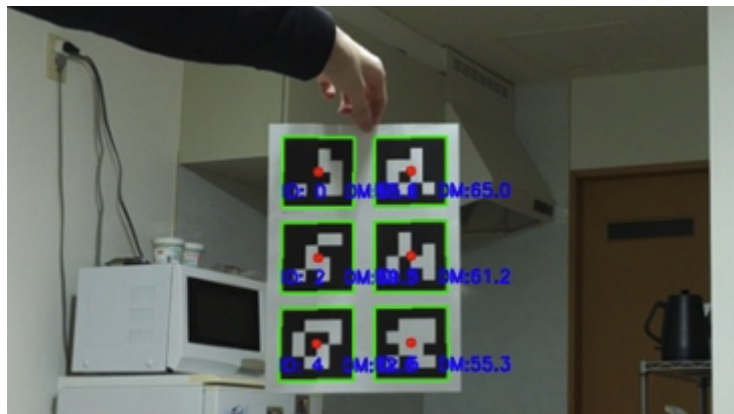


図13 Tag16h5の検出の様子

検出角度検証の結果から、マーカ―が必ずしもカメラに対し正面を向いておらずとも検出できることが分かり、台座またはマーカ―の設置位置に柔軟性が生まれたといえる。

暗所での検出の検証から、マーカ―の外周と白のグリッドを発光させることで、日射量の少ない日中の天候や夜間の場合でも検出できることがいえる。

3. トマトの成熟度モニタリングシステムの開発と調査

3.1 はじめに

本節は、画像認識技術を用いてトマトの成熟度を自動で推定するプログラムの開発プロセスと、その成果を詳述するものである。開発はGoogle Colaboratoryを環境として進め、深層学習モデルの構築には公開データセットLaboro Tomatoを使用した。モデルのアーキテクチャにはMobileNetV2を基盤とし、転移学習アプローチを採用することで効率的な学習を実現した。また、開発には主要な技術としてTensorFlow, Keras, OpenCVを活用した。最終的な実用化を視野に入れ、学習済みモデルをTensorFlow Lite形式に変換し、エッジデバイスであるRaspberry Pi上でリアルタイム推論を行うアプリケーションとして実装した。さらに、ビニールハウス内での運用を想定した実証実験として、Raspberry PiにWebカメラを接続し、市販のトマトを撮影して性能を評価した。

3.2 開発プロセス

3.2.1 データ準備と前処理

本研究で用いたLaboro Tomatoデータセットは、個々のトマトのバウンディングボックス情報を含む物体検出形式で提供されている。本研究の目的は画像分類であるため、このアノテーション情報に基づき、画像から各トマトの領域を個別に切り出す前処理を実施した。切り出された各画像は「完熟」「半熟」「未熟」のいずれかのカテゴリに分類・保存し、画像分類タスクに適した学習データセットを構築した。

3.2.2 モデル設計と学習アプローチ

高精度かつ効率的な学習を実現するため、転移学習アプローチを採用した。具体的には、ImageNetで事前学習済みのMobileNetV2をベースモデルとして採用した。その上で、トマトの3カテゴリ分類に特化した新たな出力層を追加したCNN(畳み込みニューラルネットワーク)を設計した。モデル性能を客観的かつ信頼性高く評価するため、5分割層化交差検証 (Stratified K-Fold Cross-Validation) を用いて学習と評価を行った。

3.2.3 モデル性能評価

5分割層化交差検証の結果、構築したモデルは平均正解率約86.92%を達成した。また、学習曲線の可視化から、過学習を抑制しつつ安定した学習が進行したことを確認した(図14)。一方で、精度は90%には到達せず、前処理手法の改善をはじめとする更なる精度向上が今後の課題であるとわかった。

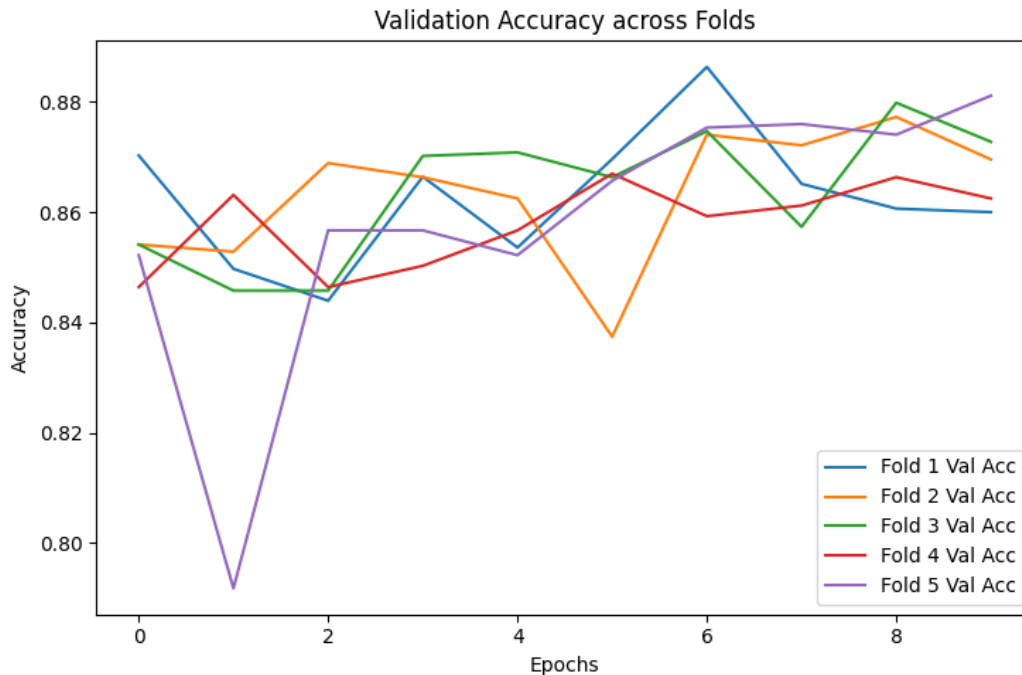


図14 5分割層化交差検証の正解率の推移

3.3 実装

実装は、モデルを生成する学習・変換プログラムと、エッジデバイス上で推論を実行するリアルタイム推論プログラムの2つから構成される。

3.3.1 学習・変換プログラム

本プログラムは、トマト成熟度分類モデルの学習と、エッジデバイス実装用のモデル変換を担う。実行環境には、Google Colaboratoryを用いた。まず、データセットをダウンロードおよび解凍し、画像とアノテーション情報(JSON形式)を読み込むことから処理が始まる。このアノテーションに基づき、個別のトマト画像を切り出し、カテゴリ別に整理された学習データを生成する。次に、メモリ使用量を抑制するため、画像データを直接メモリにロードするのではなく、ファイルパスのリストを介して学習時に逐次読み込む方式を採用している。モデル学習の段階では、MobileNetV2をベースとした転移学習を実行し、学習時には画像の回転や拡大・縮小といったデータ拡張を適用してモデルの汎化性能を向上させる。最終的に、5分割層化交差検証による性能評価を実施する。その後、全ての学習データを用いて最終モデルを再学習させた。学習済みの最終モデルは、Raspberry Piなどのエッジデバイスで高速動作が可能なTensorFlow Lite(.tflite)形式に変換して保存する。

3.3.2 リアルタイム推論プログラム

本プログラムは、Raspberry Pi上で動作し、カメラ入力からリアルタイムでトマトの成熟度を分類するアプリケーションである。プログラムは起動後、まず学習済みのTensorFlow Liteモデル(.tfliteファイル)をメモリにロードし、接続されたカメラを初期化して映像のキャプチャを開始する。メインループ内では、カメラから継続的にフレームを取得し、リアルタイムでの推論処理を繰り返す。具体的には、取得したフレームをモデルの入力仕様に合わせて160x160ピクセルへのリサイズ、および正規化といった前処理を行う。次に、前処理済みの画像データをモデルに入力し、推論を実行する。これにより、入力画像が「完熟」「半熟」「未熟」の各カテゴリに属する確率値が出力

される。最後に、最も確率の高いカテゴリを最終的な予測結果として採用し、その結果をディスプレイにリアルタイムでテキスト表示する。また、キーボードのsキーでカメラに映っているものを撮影できる(図15)。

```
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate#  
(.venv) root@raspi-yumura:/home/rainbow/tomate# python screenshot_realtime_tomato_inference.py  
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.  
カメラの起動に成功しました。  
's'キーを押すと画像を保存します。  
'q'キーを押すと終了します。  
(ウィンドウは表示されません。このターミナルを見て操作してください)  
予測: green (70.3%) (53.5%)  
画像を保存しました: captured_images/capture_20250912_105348.jpg  
予測: half_ripened (93.0%)
```

図15 Raspberry Piで動作しているリアルタイム推論プログラム

3.4 実験

3.4.1 実験方法

実験は室内にて、Raspberry PiにUSB接続のWebカメラを接続した環境で行った(図16)。



図16 Raspberry PiとWebカメラ

撮影対象には市販のトマトを3個(完熟トマト, 半熟トマトA, 半熟トマトB)を用いた(図17)。



完熟トマト



半熟トマトA



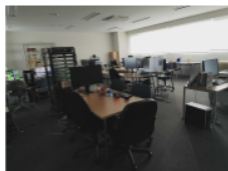
半熟トマトB

図17 市販のトマト3個

トマトの成熟度を自動で推定するプログラムには今回作成したリアルタイム推論プログラムを用いた。実験の核となる照明環境については、部屋の照明やデスクライト、窓からの自然光の組み合わせにより、5段階の明るさレベルを設定した(表1, 図18)。最も明るいレベル5から、照明を全て消したレベル1まで、段階的に照度を落とした各環境で写真を記録した。各環境の照明環境は表1に、撮影場所を図5に示す。撮影は、各明るさレベルにおいて3つのトマトをそれぞれ2回ずつ、角度を少し変えながら行い、取得した全30枚の画像データをプログラムで推論した。

表1 各環境の照明条件

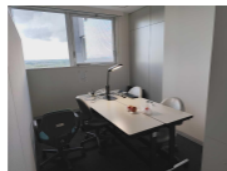
	室内灯	デスクライト	自然光
レベル1	なし	なし	なし
レベル2	なし	なし	あり
レベル3	なし	あり	あり
レベル4	あり	なし	なし
レベル5	あり	あり	なし



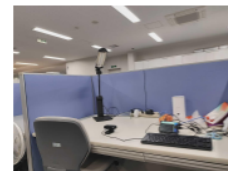
レベル1



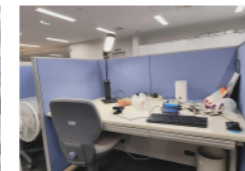
レベル2



レベル3



レベル4



レベル5

図18 各環境の撮影場所

3.4.2 実験結果

取得した全30枚のトマト画像データに対し、開発したプログラムで推論した。その結果、予測カテゴリは撮影レベルや試行回数によらず、すべての画像で「完熟」と判定した。各画像の予測カテゴリと信頼度の詳細を、それぞれ表2および表3に示す。

また、信頼度はすべての画像データで90%以上の値を示した。特に、完熟トマトの画像は、すべての撮影レベルにおいて常に99%の信頼度を示した。一方、半熟トマトAおよびBの画像では、撮影レベルによる信頼度の変動を確認した。レベル1から3では、撮影角度による信頼度の変化はほとんどなかったが、レベル4および5においては、撮影角度を変えた際に信頼度が低下する傾向を示した。

表2 予測カテゴリの一覧

		完熟トマト		半熟トマトA		半熟トマトB	
		1枚目	2枚目	1枚目	2枚目	1枚目	2枚目
予測カテゴリ	レベル1	完熟	完熟	完熟	完熟	完熟	完熟
	レベル2	完熟	完熟	完熟	完熟	完熟	完熟
	レベル3	完熟	完熟	完熟	完熟	完熟	完熟
	レベル4	完熟	完熟	完熟	完熟	完熟	完熟
	レベル5	完熟	完熟	完熟	完熟	完熟	完熟

表3 信頼度の一覧

		完熟トマト		半熟トマトA		半熟トマトB	
		1枚目	2枚目	1枚目	2枚目	1枚目	2枚目
信頼度	レベル1	99.40%	99.60%	98.80%	98.50%	98.70%	99.10%
	レベル2	99.80%	99.80%	97.30%	96.10%	96.90%	94.50%
	レベル3	99.40%	99.50%	98.40%	96.10%	96.70%	98.40%
	レベル4	99.20%	99.20%	99.30%	93.70%	90.20%	97.00%
	レベル5	99.40%	99.20%	99.00%	98.70%	94.10%	95.30%

4. おわりに

4.1 ARマーカを用いたビニールハウスの窓開閉監視システムにおける課題

本研究で作成したシステムはハウス内に設置することを想定して構築したが、実際にハウス内で運用した実績がない。また本システムはRaspberry Piがインターネットに接続できなければ、Slackに情報を送信することができない。そのためハウス内で運用するためには通信環境を構築するところから始めなければならない。

本研究ではハウスの換気窓の開閉状態をARマーカを検出することで判断し、温度情報とともにメッセージアプリに通知するシステムを構築した。また、ARマーカの有用性を確認するため検出距離、検出角度、暗所での検出に関する検証を行った。今後はハウス内に装置を設置し、長期的な運用テストを行う必要がある。

4.2 トマトの成熟度モニタリングシステムにおける課題

本実験で開発した学習済みモデルには、いくつかの興味深い特徴が確認された。学習済みモデルは、真っ赤に熟したトマトを高い精度で識別する。その識別精度は約100%であった。一方で、緑色が残る半熟トマトを完熟と誤認する課題が判明した。

この誤認は、学習済みモデルが学習過程においてトマトの赤い部分を過度に重要視した結果生じると考えられる。すなわち、学習済みモデルはわずかな赤い部分を検出すると、残りの緑色の部分を無視して完熟と判断する傾向を持つ。

しかし、学習済みモデルは半熟トマトを認識する際、その判断信頼度が完熟トマトの場合と比較して低下し、不安定になる。このことは、最終的な判断は誤っているものの、学習済みモデル内部では対象物が完熟トマトとは異なることを検知し、判断に迷いが生じている可能性を示唆する。

さらに、照明環境も学習済みモデルの判断に影響を与えた。薄暗い環境下では判断が安定する一方、過度に明るい環境下では判断が不安定になった。これは、薄暗い環境が学習済みモデルによる正確な色情報の取得を阻害するためと推測される。

以上の結果から、学習済みモデルは完熟トマトの選別に限定すれば実用可能であるといえる。しかし、多様な成長段階のトマトを正確に識別するためには、学習方法の改良が必要である。

4.3 まとめ

本調査の結果から、画像処理技術はハウス農業のスマート化に有効であることが示された。一方、今後は実環境での長期運用テストや、より多様な成長段階の作物を正確に識別するための学習モデルの改良が課題として挙げられる。