

伸縮可能なモジュール式ロボットを用いた  
トマトの病気モニタリングシステムに  
関する研究

Tomato Disease Monitoring System  
Using a Modular and Extendable Robot

欧陽 辰

Chen Ouyang

(2020 年度入学, 20834305)

指導教員 清水郁子 准教授

東京農工大学大学院 工学府 電子情報工学専攻 知能・情報工学専修

# 目次

<b>第1章 序論</b>	<b>1</b>
1.1 背景	1
1.1.1 可能な解決策	3
1.2 本研究の目的	4
1.3 本論文の構成	6
<b>第2章 関連研究と技術</b>	<b>7</b>
2.1 伸縮・昇降可能なデザイン	7
2.2 MobileNetv2	10
2.3 YOLOv5	12
2.4 関連技術とツール	14
2.4.1 Robot Operating System	14
2.4.2 データのアノテーション	14
2.4.3 ロボットの製作技術と工具	14
<b>第3章 トマトの病気モニタリングシステム</b>	<b>17</b>
3.1 トマト栽培温室	17
3.2 アーキテクチャデザイン	17
3.3 ロボットデザイン	19
3.3.1 メカニズム	20
モバイルシャーシモジュール	20
ネスタブルモジュール	24
配置	27
3.3.2 ハードウェアシステム	28
3.3.3 電気システム	30
3.3.4 ソフトウェアシステム	33
3.4 データサーバ	34
3.4.1 2段病気検出モデル	35
3.5 システムのワークフロー	36
<b>第4章 実験、結果とディスカッション</b>	<b>39</b>
4.1 実験環境の紹介とロボットの配置	39

4.2	モニタリングのための定義とセッティング	39
4.3	2 段階病気検出モデルのトレーニング	43
4.3.1	データセット	43
4.3.2	検出ネットワークのトレーニングと結果	48
4.3.3	検証ネットワークの訓練と結果	50
4.3.4	2 段階病気検出モデルの結果	51
4.4	システムの結果	53
4.5	ディスカッション	55
4.5.1	システムの実装結果について	55
4.5.2	データセットの分割とネットワークの比較	55
4.5.3	2 段階病気検出モデルについて	56
4.5.4	本研究の限界と今後の課題	58
<b>第 5 章</b>	<b>結論</b>	<b>59</b>
5.1	まとめ	59
5.2	研究の成果と貢献	59
5.2.1	新型ロボットの提案	59
5.2.2	誤検出率の減少方法の提案	59
5.2.3	トマト病気モニタリングの自動化の実現	60
5.3	他の作物への可能性の検討	60
5.4	今後の課題と展望	60
	<b>参考文献</b>	<b>62</b>
	<b>謝辞</b>	<b>68</b>
<b>付録 A</b>	<b>回路図</b>	<b>69</b>
A.1	モバイルシャーシモジュールと L1 の回路図-メインボード, レイヤ 1	70
A.2	モバイルシャーシモジュールと L1 の回路図-メインボード, レイヤ 2	72
A.3	モバイルシャーシモジュールと L1 の回路図-ドライブモジュールボード	74
A.4	L2 の回路図	76
A.5	L3 の回路図	78
<b>付録 B</b>	<b>設計図</b>	<b>80</b>
B.1	モバイルシャーシモジュール	81
B.2	L1	83
B.3	L2	85
B.4	L3	87

# 目次

1.1	温室でトマトの画像やビデオデータを収集するロボット. (a-c) 固定構造で撮影範囲も固定のロボット [23–25]; (d, e) カメラが一定範囲で移動でき, 撮影範囲が調整可能なロボット [26, 27]; (f) 複数のカメラを上げられ, 撮影範囲はほとんどのトマトをカバーしている [28]. . . . .	5
2.1	シザーズ式のデザインとその応用例. (a) シザーズ式のデザイン [32]; (b) 葉摘みロボットへの応用 [31]. . . . .	8
2.2	ワイヤーロープ&滑車式のデザインを応用した電動スタッカ [34]. . . . .	9
2.3	マスト式のデザインを持つ自走式リфта [35, 36]. . . . .	9
2.4	MobileNetv2 の「bottleneck」の構造 [38]. . . . .	11
2.5	YOLOv5l のアーキテクチャ [43]. . . . .	13
2.6	Makesense.ai の公式サイトが示す例 [46]. . . . .	15
2.7	アルミプレートにケガキ作業を行った例. . . . .	16
2.8	手作業より作成した回路基板の例. . . . .	16
3.1	トマトを栽培する温室の例. . . . .	18
3.2	トマトの病気モニタリングシステムのアーキテクチャ. . . . .	19
3.3	モバイルシャーシモジュールの実物画像 (補助サポートモジュールを除く). . . . .	21
3.4	モバイルシャーシモジュールは3つのモジュールで構成されている. (a) モバイルシャーシモジュール; (b) ベースフレームモジュール; (c) 駆動モジュール; (d) 補助サポートモジュール. . . . .	22
3.5	モバイルシャーシモジュールの移動方式の例. (a) 前進と後退; (b) 超信地旋回; (c) 横に移動; (d) 対角線に動く. . . . .	23
3.6	ホイールと温水パイプの断面図. . . . .	24
3.7	ネスタブルモジュールは3つのモジュールに分けることができる. (a) 直方体フレームモジュール; (b) カメラの昇降モジュール; (c) 牽引モジュール. . . . .	25
3.8	直方体フレームモジュールの実物画像. . . . .	26
3.9	小さい直方体フレームモジュールは大きい直方体フレームモジュールの中で上下に滑れる. . . . .	27
3.10	ロボットの配置の例. . . . .	28
3.11	本研究では作成したロボットの实物画像. (a) スタンダードモード; (b) 伸展モード. . . . .	29

3.12	ハードウェアシステムのブロック図. 3つのネスタブルモジュールが使用されることを例として示されている. . . . .	30
3.13	ロボットにインストールされているハードウェア. . . . .	31
3.14	ロボットの伸縮状態を把握するリミットスイッチ. . . . .	32
3.15	グラフィカルユーザインターフェ이스の例. ユーザはこのインターフェースを通してロボットに指示する. ボタンは実験に対応している. . . . .	32
3.16	ソフトウェアシステムの構成. 異なる色はそれぞれのデバイスにインストールされていることを意味する. . . . .	34
3.17	トマトの病気検出モデルは2段構造にデザインされている. . . . .	35
3.18	トマトの病気モニタリングシステムのワークフロー. (a) ユーザーがジョイスティックを用いてリモートでロボットを制御し, 温水パイプの起点まで移動させる; (b) 「自動収集モード」が選択されたら, ロボットが伸展モードに変形し, 温水パイプに沿って自律走行してトマトの画像データを収集する. (c) ロボットが事前に設置した位置に到達すると, スタンダード状態に戻り, そして起点に戻る. (d) 収集した画像データがサーバーに送信されてから, ロボットはそのままユーザの指示を待つ. (e) ユーザがロボットをコントロールして次の温水パイプに移動し, (b)(c)(d) が繰り返れる. (f) 最後はデータサーバで実行されている病気検出プログラムが病気のトマトの具体的な位置をユーザに通知する. . . . .	38
4.1	実験を行った温室環境. . . . .	39
4.2	実験を行う温室の概略図. ロボットはオレンジ色のポイントで自動停止してトマトの画像データを収集するように設定. . . . .	40
4.3	(上) スタンダードモードから伸展モードに変形するプロセス; (下) 3つのカメラ (赤い矢印が指しているところ) は同期に上昇し, 上昇した毎に3枚の画像データが収集される. 9回上昇したら元のポジションに戻り, ロボットは次の位置に移動する. . . . .	41
4.4	温水パイプの上に画像データを収集するフローチャート. . . . .	42
4.5	尻腐れ果の症状遷移 [56]. . . . .	44
4.6	ある場所の画像データの一部. 画像の撮る順序は現実世界の低処 (Low) と高処 (High) に対応する. . . . .	45
4.7	ラベル付けされたトマトの例. . . . .	46
4.8	データセット3の分割模式図. 場所に基づいて分割されており, トレーニング, バリデーションおよびテストに用いるデータの撮影位置が異なっている. . . . .	47
4.9	分類ネットワークをトレーニングするためのデータの例. (上) 尻腐れ果; (下) その他. . . . .	51
4.10	異なる信頼度の閾値で2段病気検出モデルと1段目のネットワーク (YOLOv5l) のみを実行した結果. (a) 偽陽性率の比較; (b) 偽陰性率の比較. . . . .	52

4.11 信頼度の閾値が0.1に設定された場合の検出結果の例. 2段病気検出モデルは青い矢印が指している誤検出を削除したが, 赤い矢印が示した誤検出は削除できなかった. . . . .	54
4.12 システムによって報告された尻腐れたトマトの分布. 赤い点は列 $x_0$ の結果で, 緑色の点は列 $x_1$ の結果である. より濃い点は複数の病気のトマトが重なっていることを意味する. (a)4月10日の結果; (b)5月12日の結果; (c)6月10日の結果. . . . .	55
4.13 提案の2段階病気検出モデルの失敗例 (赤い矢印が指しているところ). (a) 失敗の検出例; (b) 元の画像. . . . .	57

# 目次

1.1	トマトの用途による分類とその特徴 [3]. . . . .	1
1.2	トマトの一部の病気とその特徴 [1,4]. . . . .	2
1.3	温室でトマトの画像やビデオデータを収集するロボットの比較. . . . .	5
2.1	伸縮・昇降のためによく使われているベーシックなデザイン [29]. . . . .	7
2.2	MobileNetv2 のアーキテクチャ [38]. . . . .	10
2.3	ロボットを開発するためのソフトウェア. . . . .	15
2.4	主な加工用具. . . . .	16
3.1	キーの組み合わせでロボットを制御する. . . . .	36
4.1	3つの分割条件で作成したデータセットの詳細. . . . .	48
4.2	データセット1に基づいてトレーニングした結果(6:2:2の割合でランダムに分割). . . . .	49
4.3	データセット2に基づいてトレーニングした結果(日付による分割). . . . .	49
4.4	データセット3に基づいてトレーニングした結果(場所による分割). . . . .	49
4.5	用いたオーグメンテーション手法の詳細. . . . .	50
4.6	分類ネットワークをトレーニングするためのデータセットの詳細. . . . .	50
4.7	各分類ネットワークのトレーニング結果. . . . .	51
4.8	信頼度の閾値が0.1に設定された場合, 2段病気検出モデルの分類ネットワークの混同行列. . . . .	53

# 第1章 序論

## 1.1 背景

トマト（英名：Tomato，学名：*Solanum lycopersicum* Mill）はナス科に属し [1]，2021 年世界中で約 1.89 億トンのトマトが生産され [2]，世界的に見てジャガイモに次ぐ重要な野菜<sup>1</sup>である [3]。文献 [4] によれば，トマトは果実に各種の糖やペクチン質，クエン酸などの有機酸，アミノ酸，ビタミン類やカリウム，カルシウムなどのミネラルが豊富に含まれており，栄養価のバランスがとれた野菜である。

トマトにはさまざまな品種があり，用途によって分類すれば生食用品種と加工用品種に分けることができる [4]。表 1.1 に示すように，加工用トマトは露地で栽培されるが，生食用トマトは温室でも栽培される。近年では温室栽培が拡大されており，トマトは最も多く生産されている温室野菜の 1 つとなっている [5]。文献 [3] によると，日本の生産者は生食用トマトに焦点を合わせており，日本国内における生食用トマトの生産はほとんど温室で行われている。

表 1.1: トマトの用途による分類とその特徴 [3].

類別	用途	生産方式	生育特性
加工用	ケチャップなどの原料	露地	心止まり <sup>2</sup> ，草丈が低い
生食用	生食，調理	露地，温室	温室では非心止まり，誘引が必要

文献 [7] によれば，温室とは，内部の温度を一定に保てるようにし，作物が生育する環境を制御または部分的に制御する建物である。温室は作物の生育に適さない時期に生育期を与えて収量を増やすことができ，主に野菜，花きと林木などの植物栽培や育苗に使用される。温室には多くの種類があり，さまざまな屋根トラスの材料，照明材料，形状と加熱条件に応じて，温室の構造や性能も変わる。現代の温室には温度，湿度，光などの条件を制御する設備が整備され，コンピュータによる自動制御が使用され，作物が必要とする最適な環境条件を作り出せる。温室は自動的に水や肥料を撒くことができ，温度や湿度などの環境条件も制御できる。しかし，温室で栽培されてもトマトは依然として病気<sup>3</sup>にかかる。多数の病気は逆に発生しやすくなる [8]。

病気の発生はトマトの減収を招く。病気を見つければ，薬剤散布などの対処により損失を減少させることができる。しかし，病気の発見が遅れると損失が大きくなる。特に灰色かび

<sup>1</sup>場合によってはトマトは果物として認識されることもあるが，本論文ではトマトを野菜として扱う。

<sup>2</sup>茎となるべき新しい成長点が伸長せず，茎の伸長が止まってしまうことである [6]。

<sup>3</sup>簡潔のため，本論文ではトマトの病害，虫害と生理障害を区別せずに「病気」として扱う。



病のように伝染性がある病気の場合、被害部位に対してできる限り早く処置しないと、被害は広がって収量に大きな影響を与える。つまり、損失を抑えて生産量を向上するためにタイムリに病気を見つける必要がある。

一方、表 1.2 に示すように、トマトの病気は外観から弁別することができるものがほとんどである。青枯病のように発病したらトマト植物体全体が青枯れた状態になるのはもちろん、トマトの葉や茎、果実などに症状や異変が現れる病気も見た目を通して病気の種類を推定できる。根腐萎凋病のように、地上から見えない根にかかる病気も存在するが、病気が進行すると植物体が萎れるため、このような病気であっても外観から推測することは可能である。

表 1.2: トマトの一部の病気とその特徴 [1, 4].

病気	特徴
青枯病	根部が枯れて地上部の生育が完全に止まり、青枯れた状態となる。
萎凋病	植物体の下方の葉から黄化して萎れ、茎の維管束部が褐色化する。
根腐萎凋病	日中に萎れ、夜間や曇雨天の時は回復するが、やがて枯死する。茎の導管褐変は地際に止まり、上部に上がらない。根腐れが激しい。
灰色かび病	果実や花、茎に灰色のかびを生ずる。
立ち枯れ	地面に近い茎が外側から褐色となり、やがて根も腐敗する。
疫病	葉、茎、果実などに不整形の褐色の病斑を作る。
トマト黄化葉巻病	新葉の縁から退色して黄化し、葉巻症状が現れ、葉が縮れて発育が停滞する。
トマト黄色えそ病	葉に褐色のえそ斑を生じ、葉の先端から黄化して萎れる。
トマトモザイクウイルス病	葉にモザイク状の病斑を呈し、奇形葉となり生育が停滞する。
ハモグリバエ	成虫の体長は 2 mm の黄色のハエで、幼虫は葉に潜ったまま線状に食害する。
トマトサビダニ	葉につき、茎葉が褐色となり、果実がさび状になる。
裂果	果実のがくを中心にして同心円状、放射状に亀裂が入るものや、軸方向に果実が裂けるものがある。
空洞果	果実の子室内の果肉部分とゼリー部分に隙間を生じ、形がいびつになる。
尻腐れ果	果実の先端の部分が黒く変色する。
着色不良果	果実の肩の部位が黄色または黄緑色に変色する。

以上のことから、頻繁にモニタリングすることでトマトの病気をできるだけタイムリに発見することが可能であることがわかる。しかしながら、トマトの株毎に人の目視によるチェックを行い病気を見つけ出すのは時間がかかり、効率が悪いだけでなくコストも高い。特に大規模な温室で栽培される場合、全てのトマトを目視で確認するのは現実的ではない。そのため、トマトを効率的にモニタリングしてタイムリに病気を発見することが重要な課題となっている。

### 1.1.1 可能な解決策

近年、ディープラーニング技術の大幅な進化により、コンピュータビジョンに関わるタスクの効率と精度が大幅に向上している。ディープニューラルネットワーク (Deep Neural Network: DNN) は高度な抽象化により頑健な特徴表現を抽出できる。そして大量のデータを学習することで、学習時に与えられた訓練データだけに対してだけでなく、未知の新たなデータの両方をうまく予測できる汎化性を持っている。文献 [9] では、DNN が 2,032 種類の病気を 129,450 枚の画像データから学習し、21 人の認定皮膚科医と同じレベルで診断可能であることが報告されている。さらに、人間より優れたパフォーマンスを持つ DNN も存在していることが報告された [10]。

近年ではディープラーニング技術を用いて、トマトを対象とし、トマトの実やトマトの病気を、検出あるいは識別する研究が数多く行われている。例えば、Wang ら [11] は YOLOv3 という物体検出ネットワークを改良し、トマトに関する異常を検出した。同様に、YOLO シリーズを改良したトマトまたはトマトの病気の検出に関わる研究が、文献 [12,13] に報告されている。YOLO シリーズ以外に、Natarajan ら [14] は Faster R-CNN という物体検出ネットワークを利用してトマトの 4 種類の病気を検出した。文献 [15] も類似した研究が報告されている。画像からトマトやトマトの病気を検出する以外に、画像を直接分類してトマトの種類や病気を識別する研究もある。例えば、Zaki ら [16] は MobileNetv2 という分類ネットワークをファインチューニングしてトマトの 3 つの病気を検出した。Jiang ら [17] によって改良された ResNet はトマトの葉の斑点病、疫病と黄化えそ病を識別することができた。さらに、Lu ら [18] は DenseNet201 を使用してトマトの種類を分類できた。これらに類似した研究は文献 [19–21] でも行われている。これらの研究から見ると、ディープラーニング技術を用いてトマトの病気をトマトの画像データから自動的に発見することが可能であると考えられる。しかし、目視で病気かどうかを判定を行わない場合であっても、人手によりトマトの画像データをとること自体に多大な時間が掛かり、非現実的である。

自動的に画像データをとる方法として、文献 [22] では、温室に栽培されている作物を自動的にモニタリングするために、レールなどのデバイスを追加し、レール上でカメラを動かして撮影する方法を提案した。この方法は温室環境を改造するためコストが高い。さらに、広範囲に設置された多数のカメラのメンテナンス作業が必要であり、実際に運用する際には問題になると考えられる。それに比べ、最近研究が盛んに行われているロボティクス技術を利用して画像データを収集することはシンプルであり、ロボットのみをメンテナンスすれば良いのでメンテナンスコストが低いことが考えられる。

ロボティクス技術により自動的に画像データを収集する方法について具体的に述べる。カメラがついている移動ロボットを利用すればトマトの画像データを収集することができ、温室を改造する必要がなくなる。温室環境はロボットにも適している。温室では、作物は規則的に並んで植えられており、通常はまっすぐな通路または温水パイプで区切られている。すなわち、ロボットが2列の作物の間で自動的に単純な直線運動を行い、装着されているカメラを通して作物の画像またはビデオデータを収集できることを意味する。

近年ではトマト温室で働くロボットを開発または利用した研究が多い。図 1.1(a) に示すように、Zu ら [23] は温室の青いトマトの画像データを収集するために 4G カメラを搭載したロボットを開発した。Seo ら [24] は温水パイプの上で走り、トマトをモニタリングするロボットを提案した。また、トマトの実を数えるためにロボットを用いた研究もある [25]。これらの研究で使われていたロボットには共通して不十分な点がある。それは、様々なトマト植物の高さに対応できないことである。表 1.1 に示すように、温室で栽培されるトマトは非心止まりタイプで、高いところまで誘引されることが一般的である。しかし図 1.1(a), 1.1(b), 1.1(c) に示す通り、これらのロボットの機械構造は固定されており、高いところまで成長しているトマトを撮影することができない。

トマト植物の高さに対応できるロボットはある。Wspanialy ら [26] はトマトの葉を撮影するためにリニアアクチュエータ装置を搭載したモバイルプラットフォームを開発した(図 1.1(d))。その装置はカメラをさまざまな垂直位置に移動してさまざまな高さに対応できる。[27] で使用されたロボットも類似したデザインで設計されている(図 1.1(e))。これらのロボットは高さを伸展できる装置を用いて一定の高さまでのトマトの画像データを取得できるが、欠点としては撮影範囲に限界がある。

また、Fontejn ら [28] の研究では、トマトの大部分をカバーするために複数のカメラを上げることができるロボットを使用している(図 1.1(f))。しかしながら、このロボットは一体型になっているため、自由に配置することができない。表 1.3 に示すように、そのロボットは 3m までのトマトを撮影できるが、トマトが 3m まで誘引されていない温室に対して余分な設計があり、コストを重視している農業においては、コストパフォーマンスが低いと考えられる。

以上をまとめると、ロボティクス技術をうまく利用すれば温室に栽培されているトマトの画像データを自動的に獲得することができ、そしてディープラーニング技術を用いて獲得した画像データから自動的に病気を検出することができる。つまり、これらの技術を統合すれば、温室において自動的にトマトをモニタリングして病気を発見することが可能になる。

## 1.2 本研究の目的

本研究ではトマトを栽培する温室において、トマトをモニタリングすることができる新型ロボットの開発を行い、病気にかかっているトマトをタイムリに検出し、病気のトマトとその位置をユーザに明示するシステムを提案することを目的とする。このシステムを遂行するには下記の項目を達成しなければならない。

- 温室において自動的に全てのトマトをモニタリングできる適切な新型ロボットの開発。



(a)



(b)



(c)



(d)



(e)



(f)

図 1.1: 温室でトマトの画像やビデオデータを収集するロボット. (a-c) 固定構造で撮影範囲も固定のロボット [23-25]; (d, e) カメラが一定範囲で移動でき, 撮影範囲が調整可能なロボット [26,27]; (f) 複数のカメラを上げられ, 撮影範囲はほとんどのトマトをカバーしている [28].

表 1.3: 温室でトマトの画像やビデオデータを収集するロボットの比較.

参考文献	ナビゲーション方式	カメラ	可変機構	コスト可制御	撮影範囲
図 1.1(a) [23]	電磁誘導	4G ワイヤレス RGB カメラ	いいえ	いいえ	~160cm
図 1.1(b) [24]	温水パイプ	ELP 4K USB ウェブカメラ	いいえ	いいえ	未記述
図 1.1(c) [25]	QR コード	RGB カメラ	いいえ	いいえ	未記述
図 1.1(d) [26]	温水パイプ	SLR カメラ	はい	いいえ	未記述
図 1.1(e) [27]	温水パイプ	デプスカメラ	はい	いいえ	未記述
図 1.1(f) [28]	温水パイプ	デプスカメラ	はい	いいえ	~300cm

- 頑健で高精度なトマトの病気を検出するモデルの提案.
- 上記の項目を統合し, 実際に働くシステムとして温室に実装.

### 1.3 本論文の構成

以上の 1.1 節及び 1.2 節の内容を受けて, 本論文では以下のような構成をとる.

第 2 章では, 伸縮・昇降可能なデザインについて紹介し, 最近の研究においてそれらのデザインを導入したロボットについて紹介する. また, 本研究で用いたディープニューラルネットワークの構造と仕組みを説明する. 最後に本研究で使ったハードウェア面やソフトウェア面のツール, ロボットの製作方法についても紹介する.

第 3 章では, 本研究で提案したトマトの病気モニタリングシステムについての詳細を示す. トマトの栽培環境の特徴に基づいたシステムの設計方針を説明する. そしてトマトの病気のモニタリングを実現するためにロボットや病気検出モデルの必要な機能や設計方針について述べる. それを元にしてシステムのアーキテクチャについて論じる. それから, 新型ロボットと病気検出モデルの詳細をそれぞれ具体的に説明する. 最後にシステムの使い方や流れについて明示する.

第 4 章では, 実験の方法と結果を出し, それに基づいてディスカッションを行う. 実験の環境について紹介し, 実験を行うためにロボットと環境に対してのセッティングや定義について紹介する. そして一番適切なデータセットを見つけるために異なる分割方法で作ったデータセットをそれぞれ説明する. それらのデータセットに基づいた実験結果を示す. それからテストデータを用いて提案の 2 段構造の病気検出モデルの実行結果とその中の 1 段目のネットワークの実行結果と比較する. モニタリングシステムの実装結果も示す. 最後に, それらの結果に対して考察を行い, 実装結果や, 異なる分割方法によって作成したデータセットの影響, 2 段構造の病気検出モデルの性能, システムの限界と不足について論じる.

第 5 章では, 本研究の内容についてまとめる. そして本研究の成果や貢献, 汎用性を言及し, 最後は今後の展望に関して述べる.

また, 付録に新型ロボットの回路図と設計図を添付する.

## 第2章 関連研究と技術

本章では、本研究と関わる伸縮・昇降可能なデザインについて概説する。そして、本研究で導入したネットワークの構造や仕組みに関して紹介する。最後は本研究で開発した新型ロボットを作るために必要なハードウェア面とソフトウェア面のツールを紹介しつつロボットの製作技術について説明を行う。

### 2.1 伸縮・昇降可能なデザイン

伸縮可能、あるいは昇降可能なデザインの主な目的は、人またはものをさまざまな高さまで上げたり下げたりすることである。世の中では伸縮または昇降可能なデザインは極めて多いため、本節では、よく見られるベーシックなデザイン<sup>4</sup>のみを挙げ、そしてそれらを組み合わせた例について説明する。

文献 [29] ではよく使用されている伸縮または昇降可能なデザインをまとめている。表 2.1 に示すように、ベーシックなデザインは主に5つがあり、それぞれにメリットと欠点がある。

表 2.1: 伸縮・昇降のためによく使われているベーシックなデザイン [29].

方式	メインのメリット	メインのデメリット
ラック&ピニオン式	高速，同期性が良い	高価
プランジャ式	機構が簡単，確実に機能	同期性が悪い，エネルギーの消費が多い
シザーズ式	コンパクトな構造，大きな積載量	均一な速度制御が困難，ストロークが短い
送りネジ式	コンパクトな構造，安定，高精度	すべりねじの場合，低速，磨りへりやすい
ワイヤーロープ&滑車式	配置しやすい，モーションがスムーズ	荷重の影響を受けやすい，慣性大きい
チェーン&sprocket式	シンプルな構造，低コスト	ワイヤーロープ&滑車式と比較すれば安定性が悪い

ラック&ピニオン式のデザインにはラックとピニオンが含まれており、ピニオンとラック

<sup>4</sup>本論文では、「ベーシックなデザイン」は「基本的に単一な方式のみで機能を実現できるデザイン」と定義する。

の組み合わせは、回転運動を直線運動に変換することで伸縮または昇降を実現する。プランジャ式のデザインは通常、油圧装置と連動して、シリンダにオイルが圧入されてプランジャを動かし、そしてプランジャが直接プラットフォームなどに作用することで伸縮または昇降を実現する。シザーズ式のデザインは日常生活によく見られる（図 2.1(a)）。このデザインはパンタアームを基本単位で構成され、パンタアームを開閉することで伸縮または昇降を実現する。ストロークは短いが、パンタアームを積み重ねて高い上昇位置を確保できる。シザーズ式のデザインはシザーリフトに応用されるのが一般的であるが、近年ではロボットのデザインにも応用されている。例えば、Xu ら [30] はシザーズ式のデザインを廃液バケツを搬送するロボットに設置することで、廃液の漏れがなくなってより安全な廃液バケツ搬送を実現した。文献 [31] ではシザーズ式のデザインを葉摘み装置を上げるニーズがあるロボットに応用して葉を収集した（図 2.1(b)）。

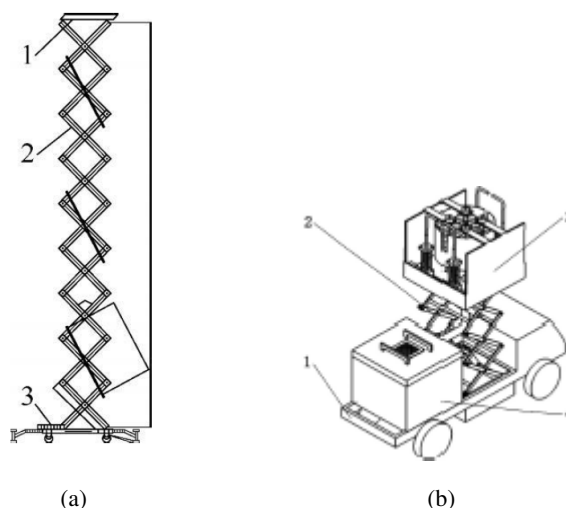


図 2.1: シザーズ式のデザインとその応用例. (a) シザーズ式のデザイン [32]; (b) 葉摘みロボットへの応用 [31].

精度が求められる場合、送りねじ式のデザインが選択肢になる。送りねじ式のデザインはモータの回転運動により、ネジ棒につけられているプラットフォームなどを移動させることで伸縮または昇降を実現する。精度が高いという特徴に基づき、張ら [33] は送りねじ式のデザインを研究に応用し、図書館の倉庫において本を管理するロボットを提案した。

日常生活でよく見られるのはワイヤーロープ&滑車式のデザインである。ワイヤーロープを牽引する方法によって仕組みは変わるが、ドラムとワイヤーロープ、滑車などを備え、駆動装置でドラムを回転させ、ワイヤーロープを巻き込むことでプラットフォームなどを昇降させるデザインは工事現場で多く見られる。最近の研究で王ら [34] はワイヤーロープ&滑車式のデザインを持つ倉庫用電動スタッカを提案した（図 2.2）。チェーン&スプロケット式のデザインの場合、一般的にモータのスプロケットを駆動機構として使用し、カップリング、伝動軸、スプロケットなどの一連の伝動で伸縮または昇降を実現する。

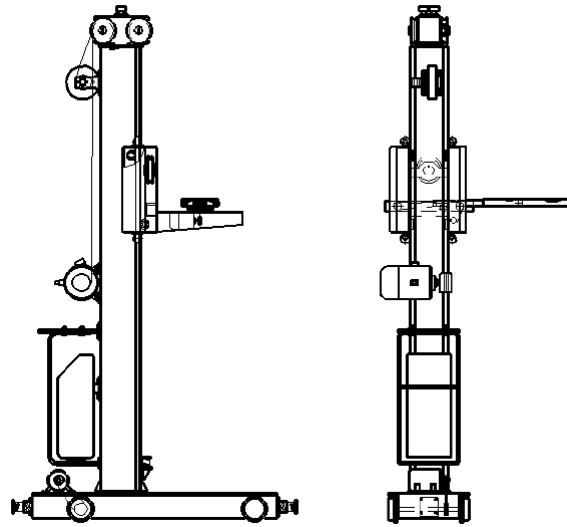


図 2.2: ワイヤロープ&滑車式のデザインを応用した電動スタッカ [34].

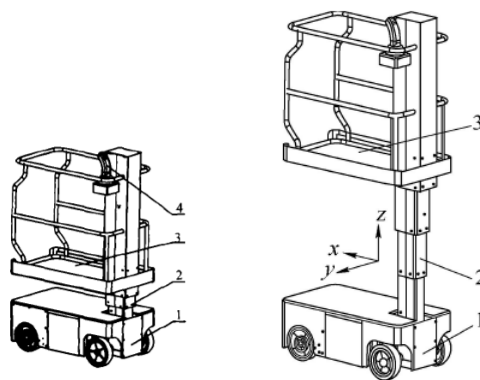


図 2.3: マスト式のデザインを持つ自走式リフタ [35,36].



実際、サイズや安定性などの需要に対して、伸縮または昇降を実現するベーシックなデザインを組み合わせたデザインはある。例として、図 2.3 の示すように、辛ら [35,36] が開発した自走式リフタはマスト式のデザインを使っている。文献の記述によると、マストの内部にはプランジャ式のデザインとチェーン&スプロケット式のデザインが組み合わせられている。マスト式のデザインのメリットとしては、高い剛性や優れた安定性、定常状態のサイズと断面積が小さいなどが挙げられる。

## 2.2 MobileNetv2

従来の畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) は計算リソースが多く必要のため、モバイルデバイスや組み込みデバイスに実装するのが不向きである。それに対して、MobileNet シリーズのネットワークは軽量に設計され、パラメータ数や計算コスト、精度などのバランスが配慮されている。MobileNet シリーズに MobileNetv1 [37] と MobileNetv2 [38], MobileNetv3 [39] という 3 つのバージョンが存在しているが、本節では、本研究に導入している MobileNetv2 のみについて紹介する。

Sandler らは 2018 年に MobileNetv2 を提案し、アーキテクチャは表 2.2 に示すように設計されている。この中、 $t$  は展開率 (次元を増やす倍数)、 $c$  は出力特徴マップのチャンネル数、 $n$  は繰り返し回数、 $s$  はストライドを意味する。

表 2.2: MobileNetv2 のアーキテクチャ [38].

Input	Operator	$t$	$c$	$n$	$s$
$224^2 \times 3$	conv2d	—	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	—	1280	1	1
$7^2 \times 1280$	avgpool 7×7	—	—	1	—
$1 \times 1 \times 1280$	conv2d 1×1	—	k	—	—

具体的に言えば、最初の入力は 3 つのチャンネルを持つ RGB 画像で、サイズが  $224 \times 224$  である。「conv2d」は標準的な畳み込みを意味し、つまり入力した画像は、ストライドが 2 で 1 回標準的な畳み込み処理をされる。そして出力特徴マップのチャンネル数は 32 になる。

「bottleneck」の構造は図 2.4 の示すようになっている。ストライドの値によって形は変わる。入力データの情報の損失をできるだけ防ぐために、最初の「Conv 1×1, Relu6」は入力

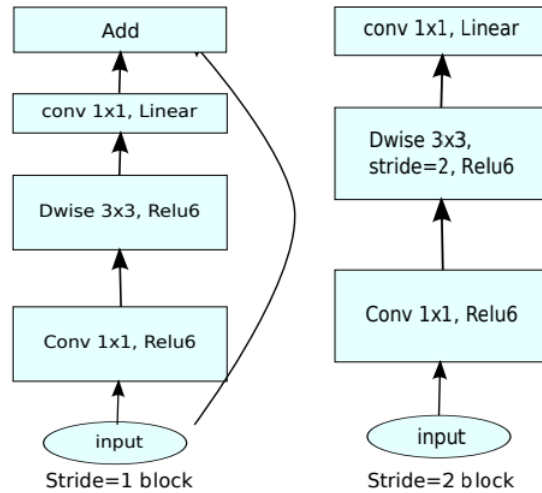


図 2.4: MobileNetV2 の「bottleneck」の構造 [38].

データの次元を増やし、表 2.2 の  $t$  で増やす倍数を調整する。「Relu6 (Rectified Linear Unit)」という活性化関数は低精度計算で使用した場合のロバスト性が高いために使われ、式は次の通りとなる。

$$\text{ReLU6}(x) = \min(\max(0, x), 6) \quad (2.1)$$

そして、「Dwise 3×3」と「conv 1×1」の連用は Depthwise Separable Convolution と呼ばれており、Depthwise Convolution (「Dwise 3×3」) と Pointwise Convolution (「conv 1×1」) の組み合わせで構成される。Depthwise Convolution は入力データをチャンネル毎にフィルタリングし、入力データのチャンネル数=フィルタ<sup>5</sup>の数=出力特徴マップの数という関係は成り立つ。入力チャンネルごとに単一のフィルタを適用することにより、軽量のフィルタリングを実行し計算コストとパラメータ数を大幅に減少できる。Pointwise Convolution は標準的な畳み込みと同様であり、フィルタのサイズは 1×1 に設計され、Depthwise Convolution の出力の線形結合を計算するだけでなく、出力の次元も減らす。ストライドの値が 1 の場合、入力と出力を直接つなぐショートカットを使用し、全体的に見ると、データの次元は低→高→低になっており、従来の Residual 構造と反対するため、Inverted Residual と命名されている。ストライドの値が 2 の場合、入力のサイズと出力のサイズは異なるため、ショートカットを適用しない。また、原文では、低次元のデータには ReLU を利用すると、多くの情報が失われることを実験で検証した。そのため、最終的な出力には ReLU6 を除外し、原文では Linear Bottleneck と命名している。

$n$  は繰り返し回数を表すため、「bottleneck」は 17 個<sup>6</sup>が設置されている。注意点として、繰

<sup>5</sup> 「カーネル」とも呼ばれるが、本論文では「フィルタ」を使う。

<sup>6</sup> 原文では 19 個と記述しているが、書き間違いであると考ええる。

り返しの場合、 $s$ の値は最初の「bottleneck」に適用し、その後の「bottleneck」のストライドは1になる。全ての「bottleneck」の後、標準的畳み込みと Global Average Pooling が行われ、最後の  $k$  は分類すべきクラスの数である。

性能実証面では、MobileNetv2 はサイズが小さいものの、精度が高く、高速に出力することがあきらかになった。

## 2.3 YOLOv5

YOLO (You Only Look Once) シリーズは YOLOv1 [40] から YOLOv8 [41] まで進化してきた。本研究は YOLOv5 [42] を導入しているため、本節では YOLOv5 について説明する。YOLOv5 には YOLOv5n と YOLOv5s, YOLOv5m, YOLOv5l などのバージョンがあるが、本研究では YOLOv5l を用いているため、YOLOv5l の構造を中心に紹介する。

YOLOv5l の構造<sup>7</sup>は図 2.5 に示す。入力は  $640 \times 640$  の 3 つチャンネルを持つ RGB 画像である。最初のレイヤ、「ConvBNSiLU」の中、「Conv」は畳み込み、「BN」は Batch Normalization, 「SiLU」は活性化関数 SiLU (Sigmoid Linear Unit) を意味する。SiLU は、関数への入力値が 0 以下の場合には出力値がほぼ 0, 入力値が 0 より上の場合には出力値が入力値とほぼ同じ値となる関数である。式は次の通りになる。

$$\text{SiLU}(x) = x * \text{sigmoid}(x) \quad (2.2)$$

そして、 $k$  はフィルタ (カーネル),  $s$  はストライド,  $p$  はパディング,  $c$  は出力のチャンネルを意味し、後ろについている値はサイズや数を表す。つまり、「k6, s2, p2, c64」はフィルタのサイズが  $6 \times 6$ , ストライドが 2, パディングが 2, 出力のチャンネル数が 64 であることを意味する。このレイヤによって、出力特徴マップは  $320 \times 320 \times 64$  になる。

重複しているレイヤの説明は省略し、続いて「C3」と「SPPF」について説明する。「C3」に 3 つの畳み込みレイヤといくつかの「BottleNeck」が含まれている。「BottleNeck1」は Residual 構造で、「BottleNeck2」は普通の畳み込み処理を 2 回行う。「SPPF」では、入力データは畳み込みレイヤを通し、出力が次の最大値プーリングレイヤ («Maxpool2d») に入力される。最大値プーリングレイヤの出力は次の最大値プーリングレイヤに入力される。最後は 4 つのレイヤの出力を融合し、畳み込みされる。このデザインは効率が良くて計算コストが少なく、ある程度にマルチスケール性能を向上している。

最後の出力は 3 つがある (緑の部分)。それぞれ違うスケールのオブジェクトに対応する。「 $80 \times 80 \times 256$ 」が入力されるレイヤは小さいなオブジェクト, 「 $40 \times 40 \times 512$ 」が入力されるレイヤは中等サイズのオブジェクト, 「 $20 \times 20 \times 1024$ 」が入力されるレイヤは大きなオブジェクトを予測する。

Loss の計算は 3 つの部分で構成されている。loss.py ファイルのソースコードから、式は次の通りになる。

$$\text{Loss} = \lambda_1 L_{\text{box}} + \lambda_2 L_{\text{obj}} + \lambda_3 L_{\text{cls}} \quad (2.3)$$

<sup>7</sup>YOLOv5 は論文がない。引用のアーキテクチャ (v6.0/6.1) は YOLOv5 の作者が認めている。

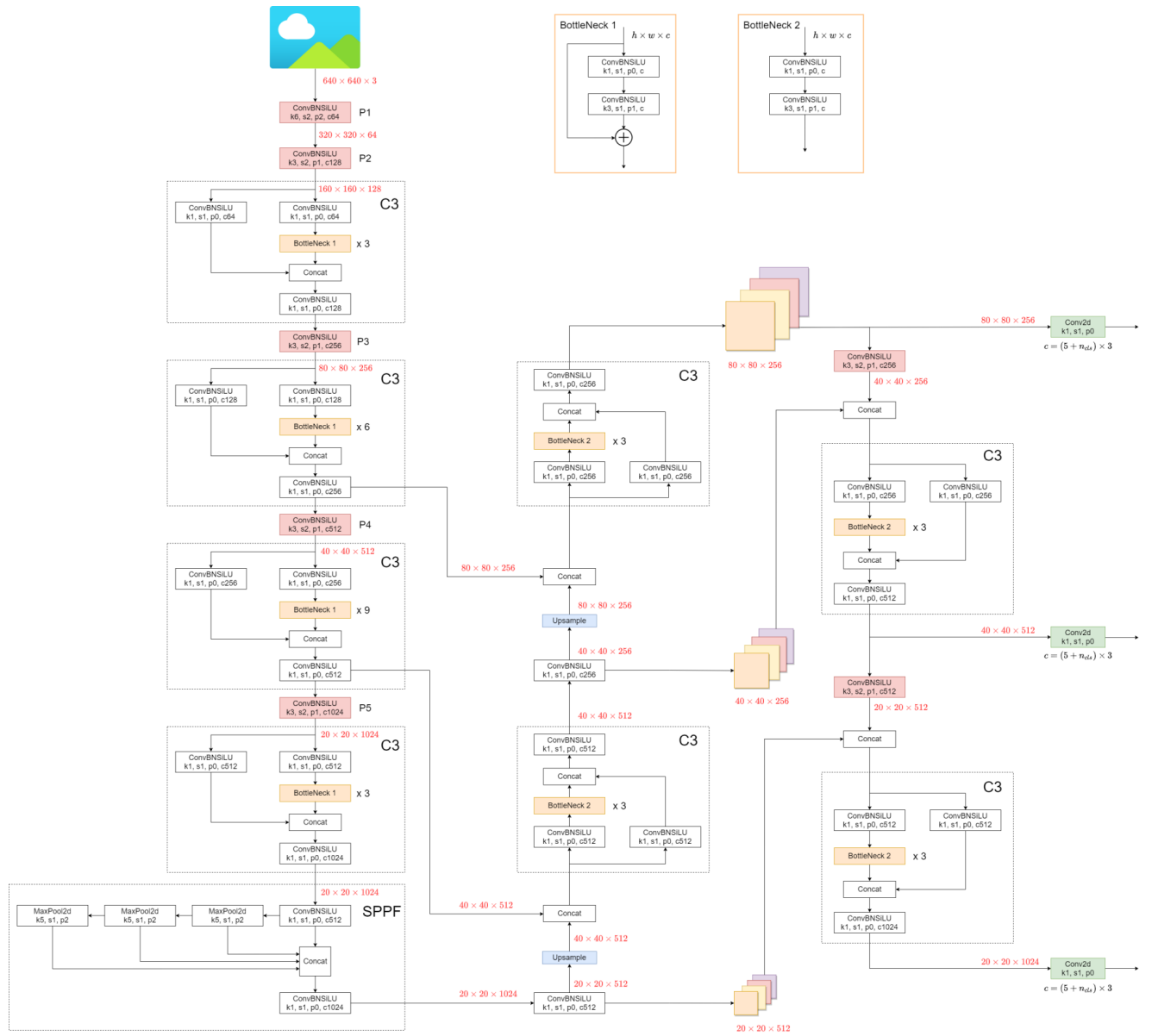


図 2.5: YOLOv5I のアーキテクチャ [43].

ここで、 $\lambda$  はバランス係数で、 $L_{cls}$  は Classification Loss,  $L_{obj}$  は Objectness Loss,  $L_{box}$  は Bounding Box Regression Loss を意味する。また、YOLOv5 はトレーニングに対していろいろ策を用いている。例えば、Mosaic と Copy Paste, Random Affine などデータ拡張の手法や、Multi-scale Training と AutoAnchor などトレーニングの手法がある。学習の時に状況に対してそれらの手法を自由に組み合わせられる。

## 2.4 関連技術とツール

### 2.4.1 Robot Operating System

Robot Operating System (ROS) [44] は、ロボットの開発のために必要な一連のライブラリとツール群を含む統合的ソフトウェアプラットフォームである。ROS の特徴をシンプルに言えば、トピックと呼ぶ仕組みを使い配信・購読型の非同期通信によりメッセージを交換するフレームワークである [45]。分かりやすい例を挙げると、ロボットに装着しているセンサが読み取った値はそのまま配信先を知らずに配信することができ、他のプログラムでは配信しているデータを購読すれば処理できる。これは非常に便利な仕組みになっており、さらに単一のデバイスだけではなく、複数のデバイスの間での通信も可能である。そして活用によって、ロボットのみではなく、複雑なシステムにも適している。

### 2.4.2 データのアノテーション

教師あり学習の場合、ディープラーニングモデルの学習に教師データが必要になる。自分のデータをネットワークにゼロから学習や転移学習させる場合、教師データは自ら作成必要がある。言い換えれば、生データにアノテーション作業を行わなければならない。図 2.6 に示すように、Makesense.ai [46] はインストールせずにブラウザのみで画像データにアノテーション作業でき、複数なフォーマットで書き出せるソフトウェア（ウェブサイト）である。例えば、YOLO をトレーニングするための教師データをアノテーションする場合、手作業で画像に含まれているターゲットを 1 つ 1 つの枠で囲んで、定義したクラスのラベルを付ける。このとき、アノテーションしたデータにはネットワークに学習させたいターゲットの位置と種類が含まれている。

### 2.4.3 ロボットの製作技術と工具

将来に機能の追加などを含めて配慮し、そして素早くプロトタイプのロボットを実現するために、本研究では筆者は自ら部品を加工して新型ロボットを作成した。

ロボットを作成するのに使っていたソフトウェアと工具は表 2.3 と 2.4 に示す。ロボットを製作するには少なくとも 3 つのステップが必要となる。具体的に、まずは Fusion 360 [47] という CAD や CAM などの機能が統合されたソフトウェアを使用し設計図を描く。Fusion 360 は 2D 図面を生成できるだけでなく、3D プリンタと CNC 用データの出力も可能である。そ

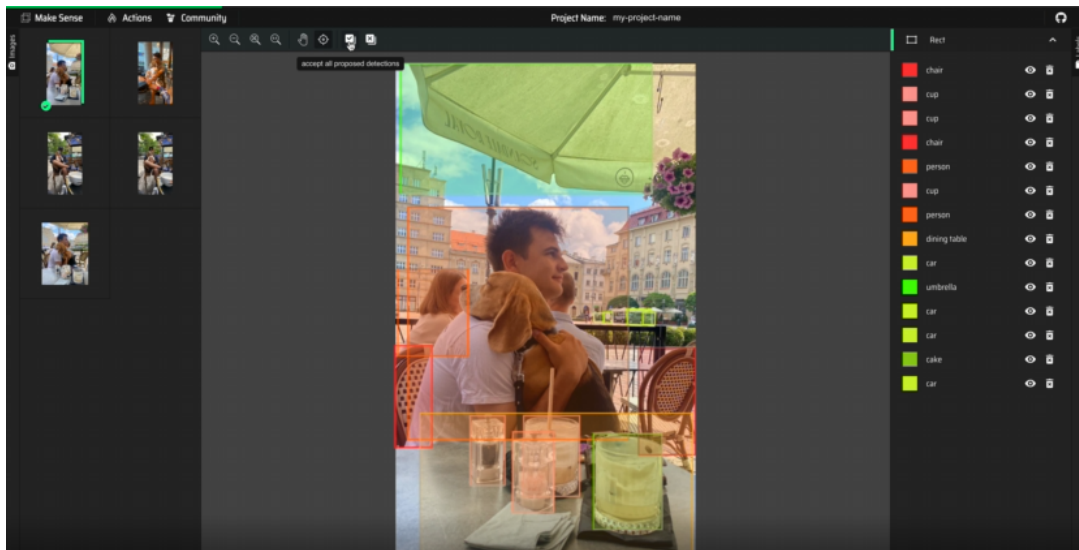


図 2.6: Makesense.ai の公式サイトが示す例 [46].

して 2D 図面の通りにハイドゲージを利用し、材料の表面に寸法の通りにケガキ作業をして穴の位置を決める (図 2.7). 穴の位置が決めたら、ボール盤を用いて穴あけ作業を行う. 円板の場合、ハイドゲージで位置をケガキするのが難しいため、CNC より加工する. 部品を 3D プリンタで作成する場合、Ultimaker Cura [48] を使い、Fusion 360 から出力したデータを 3D プリンタ用データに転換する. また、ネジで接続が難しい場合、金属の接続は電気溶接機を用い、切断は切断機を利用する.

表 2.3: ロボットを開発するためのソフトウェア.

ソフトウェア	用途
Fusion 360	3D モデリング, 2D 図面の生成
EasyEDA	回路図の作成
Ultimaker Cura	3D プリンタ用データの生成

回路図をデザインするために EasyEDA [49] を用いる. そして回路図の通りに半田付け作業を行い、回路を作成する (図 2.8). また、温室で働くため、大事なこととして防湿防錆のために回路基板に絶縁剤を塗布しておく.

最後のステップとして、当初の設計の通りに作成または加工した部品を組み立てる. 組み立て済みのロボットを動かすためにはソフトウェアシステムなどを設計して実装する必要があるが、詳細については第 3 章で説明する.

表 2.4: 主な加工用具.

ツール名	用途
ハイドゲージ	ケガキ作業
ボール盤	穴あけ
電気溶接機	金属の接続
切断機	金属の切断
はんだごてセット	回路の作成
3D プリンタ	部品の作成
CNC	穴あけ

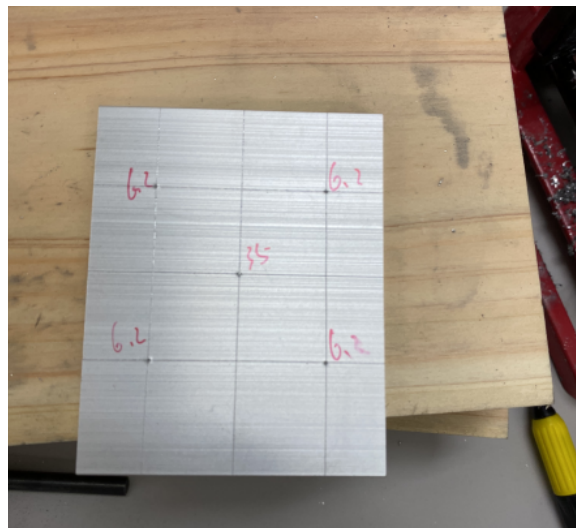


図 2.7: アルミプレートにケガキ作業を行った例.

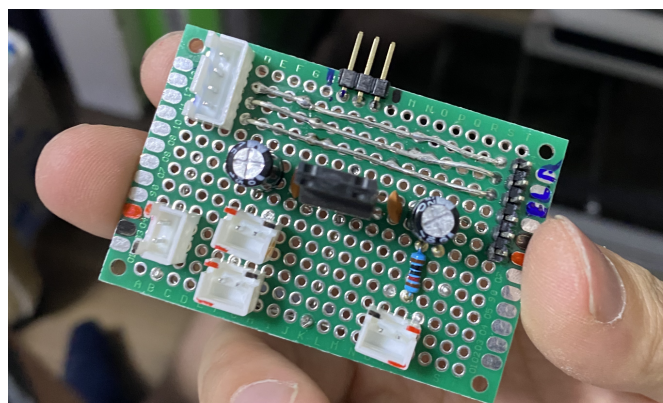


図 2.8: 手作業より作成した回路基板の例.

## 第3章 トマトの病気モニタリングシステム

本研究の目的は、ロボットを用いて温室内の全てのトマトを定期的にモニタリングし、トマトの病気の発生をタイムリに発見して、病気にかかっているトマトとその位置をユーザに明示するシステムを開発することである。この目的を達成するために、システムをどのように設計するのか、ロボットをどんな形にデザインするのかというようないくつかの問題が生じる。本章では、トマトを栽培する温室環境などについてまず紹介する。そしてそれに基づいて開発コンセプトについて説明しながら、システムのアーキテクチャのデザインを紹介し、システムの構成部分としてロボットと病気検出モデルのデザインについて詳しく説明する。

### 3.1 トマト栽培温室

図 3.1 に示すように、トマトを栽培する温室では、一般に、作物は規則的な列に植えられる。列と列の間は平らな表面または金属の温水パイプがインストールされていることが多い。このような形の温室は一般的であり、世界中の様々な地域で多く見られている。また、トマト温室の場合、栽培用のロープなどを用いて茎が高いところまでガイドされることが一般的である。ただし、その高さは一定ではなく、環境により異なる。このことから、モニタリングできる作物の高さに応じてロボットを必要十分な高さまでがモニタリングできるように設計することができるかと効率が良いということがわかる。

### 3.2 アーキテクチャデザイン

温室内で病気のトマトを発見するためのモニタリングを実現する場合、温室内に多数のカメラを設置し、環境を智能化するというアプローチも考えられる。しかし、トマトの温室は広大な面積であるものも多く、全ての作物をモニタリングできるためのカメラを設置すると、設置コストもメンテナンスコストも高い。一方で移動型のロボットを利用して温室を回って全てのトマトの画像を撮る場合には、温室の広さに関わらず移動ロボット1台で全面積をカバーすることが可能であり、またメンテナンスもそのロボットに対して行うだけで良いため簡単である。そのため、本研究では、移動ロボットを利用してトマトの画像データを収集する方法を採用する。そして収集した画像データを病気検出モデルで病気にかかったトマトを検出する。

画像データはロボット自身で処理することも可能であるが、データサーバを設置した方がよいと著者は考える。データサーバを設置するメリットはいくつかある。まず、ハードディ





British Columbia,  
Canada [50]



Tokyo, Japan [51]



Almería Province, Spain [52]



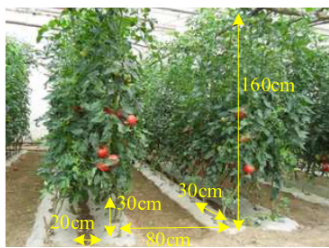
Shandong, China [53]



Beijing, China [54]



Korea [24]



China [55]



Arizona, USA [50]



North Carolina, USA [50]

図 3.1: トマトを栽培する温室の例.

スクを容易に拡張でき、大量の画像データを保存できる。次に、遠隔で病気の検出結果を確認することができる。ユーザはどこにいてもデータサーバに接続することが可能で、ロボットのオンオフ状態に関わらず、システムの状態や病気のトマトなどをいつでも確認することができる。また、グラフィックカードをインストールすることでディープラーニングモデルも実行できる。

当然ながら、ロボットとデータサーバを繋げるためにルータが必須となり、ローカルエリアネットワークを通して画像データを転送する。

まとめると、トマトの病気モニタリングシステムを実現するには移動できるロボット、画像データを保存し病気を検出するプログラムを実行するデータサーバ、およびルータ、総計3つの要素が必須となる。したがって、提案のシステムは図3.2に示すようにデザインされている。ロボットは温室内でトマトの画像データを収集するタスクを実行する。収集された画像データはルータを経由しデータサーバに転送される。データサーバは画像データを保存するだけでなく、トマトにかかっている病気を検出するモデルを実行する。

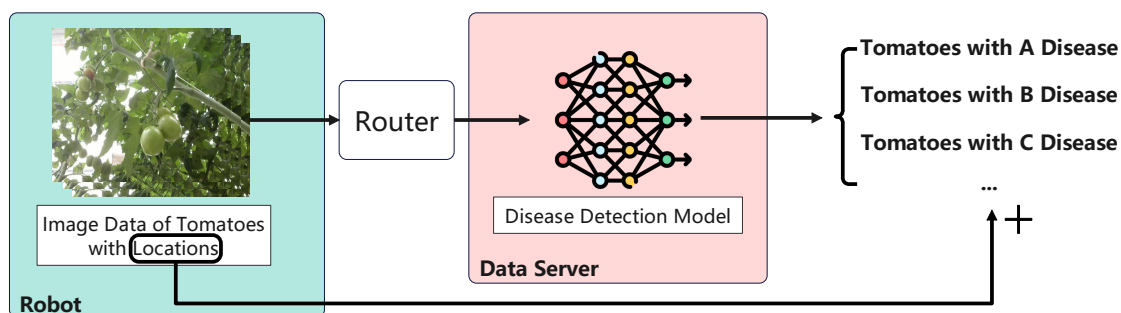


図 3.2: トマトの病気モニタリングシステムのアーキテクチャ。

詳しく述べる、まずロボットは温室を一周して画像データを収集する。画像データ収集の際には、各画像にロボットの位置に基づいた位置情報を付与する。そして病気のトマトを含む画像データはローカルエリアネットワークを通してデータサーバまで送信される。データサーバは病気検出モデルが含まれているプログラムを実行し、画像データから病気にかかっているトマトを検出する。元の画像の位置情報を加えて、最終的にシステムは病気の種類を含めて、病気にかかっているトマトとそれらが温室における具体的な位置を出力する。この出力を通して、ユーザは病気にかかったトマトに対して早期に対処することができる。

### 3.3 ロボットデザイン

本研究で提案するロボットは、栽培されるトマトと温室の特徴に基づいてデザインされている。それらの特徴については3.1節で述べたが、本節では前述した特徴に基づきロボットデザインを説明する。まずはロボットのメカニズムについて説明する。そして、電気システムに関して述べる。その後は制御ハードウェアについて説明し、最後はソフトウェアについて説明する。

また、このロボットは主に軽量で加工することが容易なアルミフレーム、アルミプレート、アクリルプレート、および3Dプリンタ部品で構成されている。付録に提案のロボットの回路図と設計図を添付している。

### 3.3.1 メカニズム

提案のロボットはモジュール式に設計されている。1つのモバイルシャーシモジュール (Mobile Chassis Module) と複数のネスタブルモジュール (Nestable Module) から構成されている。小さなネスタブルモジュールは大きなネスタブルモジュールに入れることができる。一番大きなネスタブルモジュールはモバイルシャーシモジュールに取り付ける。小さいネスタブルモジュールは、それよりも大きいネスタブルモジュールの中で上下に滑ることができる。モジュール式にデザインする理由およびメリットとして主に下記の3つが挙げられる。

- 3.1節で紹介したように、トマトの茎は栽培用ロープによってさまざまな高さにガイドされるため、撮影するためにロボットはさまざまな高さ対応する必要がある。
- 固定式のロボットはあらかじめ決められている高さには対応できるが、汎用性が低い。モジュール式のデザインは状況に応じて自由に配置することが可能で、コストもコントロールできる。
- ロボットを現場まで搬送するのに便利であり、モジュールに分割してメンテナンスできるためメンテナンス作業も容易である。

本項では、モバイルシャーシモジュールとネスタブルモジュールについて詳しく説明する。

#### モバイルシャーシモジュール

モバイルシャーシモジュールはさらにベースフレームモジュール (Base Frame Module)、駆動モジュール (Drive Module) と補助サポートモジュール (Auxiliary Support Module) に分けることができる。

ベースフレームモジュールはベースのフレームであり、断面20×20mmのアルミフレームによって構成されている。図3.3に示す画像のように、ベースフレームモジュールの中心と周りにはアクリルプレートがインストールされ、制御用コンピュータやモータドライバ、回路基板などはアクリルプレートに固定されている。

温室では、人がいるだけでなく、たくさんの機器も働く。つまり、人や機器と同じ環境で働くため、ロボットの移動には柔軟性が必要になる。そのため、移動機能を担当するモバイルシャーシモジュールは四輪駆動に設計されており、4つの駆動モジュール (図3.4(c)) によって移動する。図3.4に示すように、サーボモータ、直流ギアモータ、3Dプリンタホイール、アルミ円板、アルミアングル、フランジと角パイプで構成されている。これらのうち、3Dプリンタホイールは環境により適したものをする。これは、3.1節で述べたように、温室環境

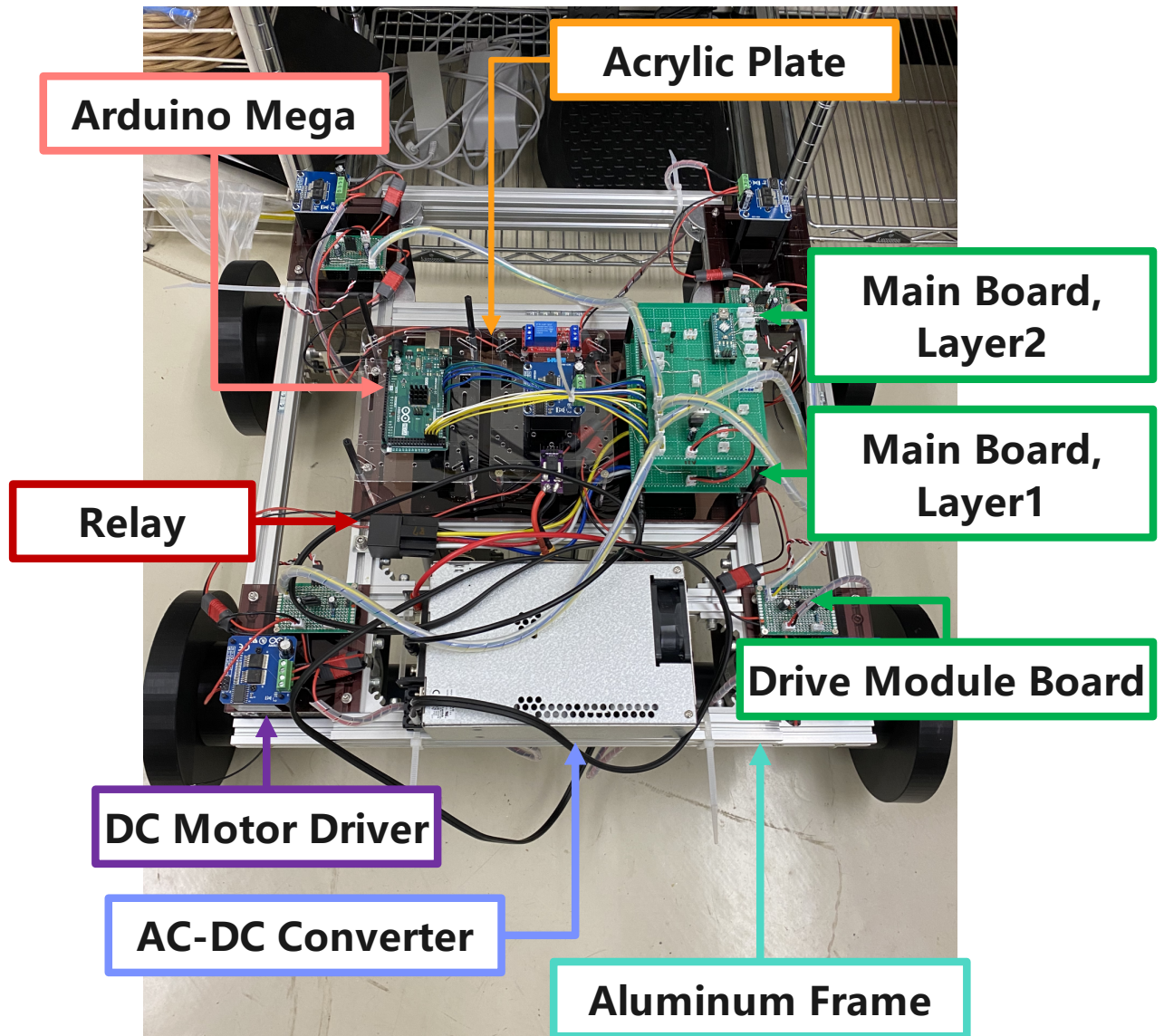


図 3.3: モバイルシャーシモジュールの実物画像 (補助サポートモジュールを除く).

において、ロボットはコンクリート、土、温水パイプなど異なる環境を走る可能性がある。しかしこの3つの条件を同時に満たすホイールのコストは高いため、駆動モジュールのホイールは容易に取り外しができるように設計されている。この設計により、実際の環境に応じて適切なホイールを選択して装着する。なお、本研究で実験を行った温室では、コンクリートおよび温水パイプ上を走行する。コンクリートおよび温水パイプ上を走行可能なホイールを低コストで作成するために、3Dプリンタを用いている。

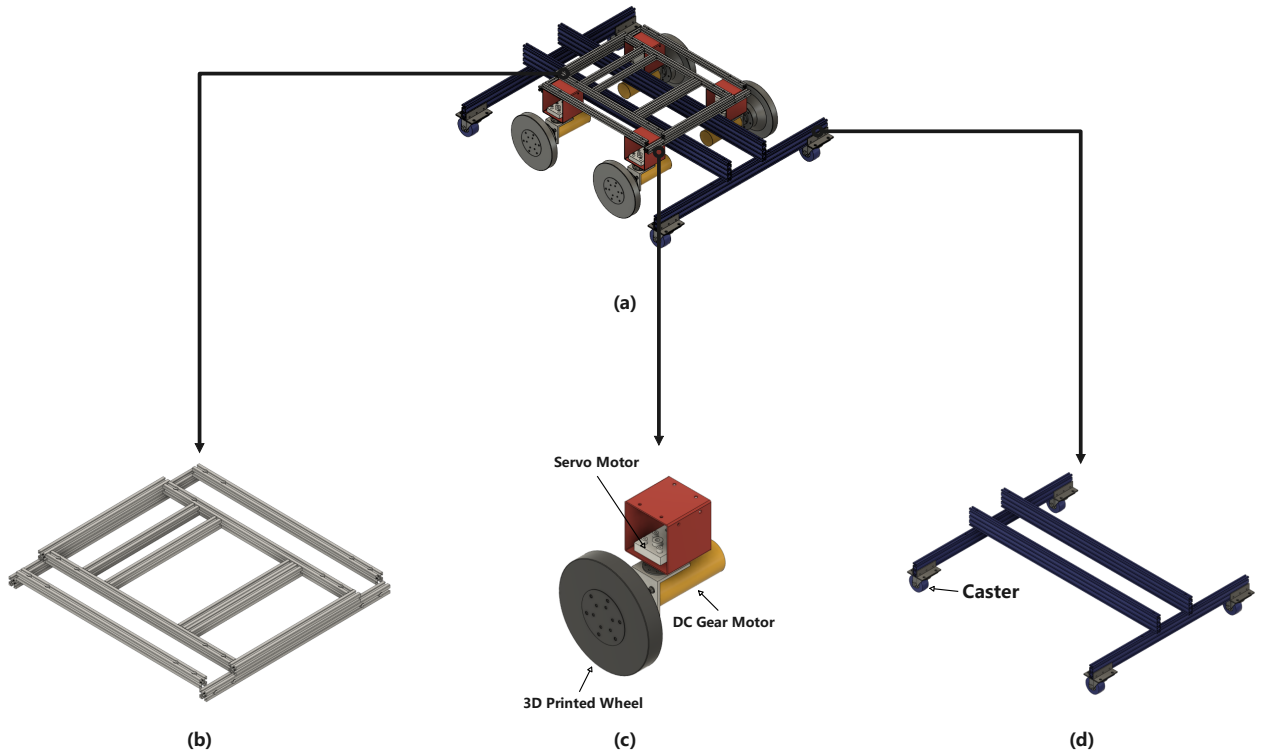


図 3.4: モバイルシャーシモジュールは3つのモジュールで構成されている。(a) モバイルシャーシモジュール；(b) ベースフレームモジュール；(c) 駆動モジュール；(d) 補助サポートモジュール。

サーボモータはロボットの進行方向を変えることができ、直流ギアモータはロボットの移動の動力を提供する。駆動モジュールはベースフレームモジュールの四角に取り付ける。図 3.5 に示すように、4つの駆動モジュールを使用することで、ロボットは前進と後退だけでなく、サーボモータの回転により超信地旋回することや横に移動することなど、全方向に移動することができる。当然ながら、サーボモータを固定し、差動で動くことに設定することも可能である。具体的な動き方は場合次第となる。このような機構により、ロボットは十分なフレキシビリティを持っていると考える。また、ホイールは3Dプリンタにより作られているため、2つのアルミ円板をホイールの両側に追加し、取り付ける時に凹みを避ける。

ホイールは3Dプリンタにより作られているため、ロボットの重さを長く耐えることが難

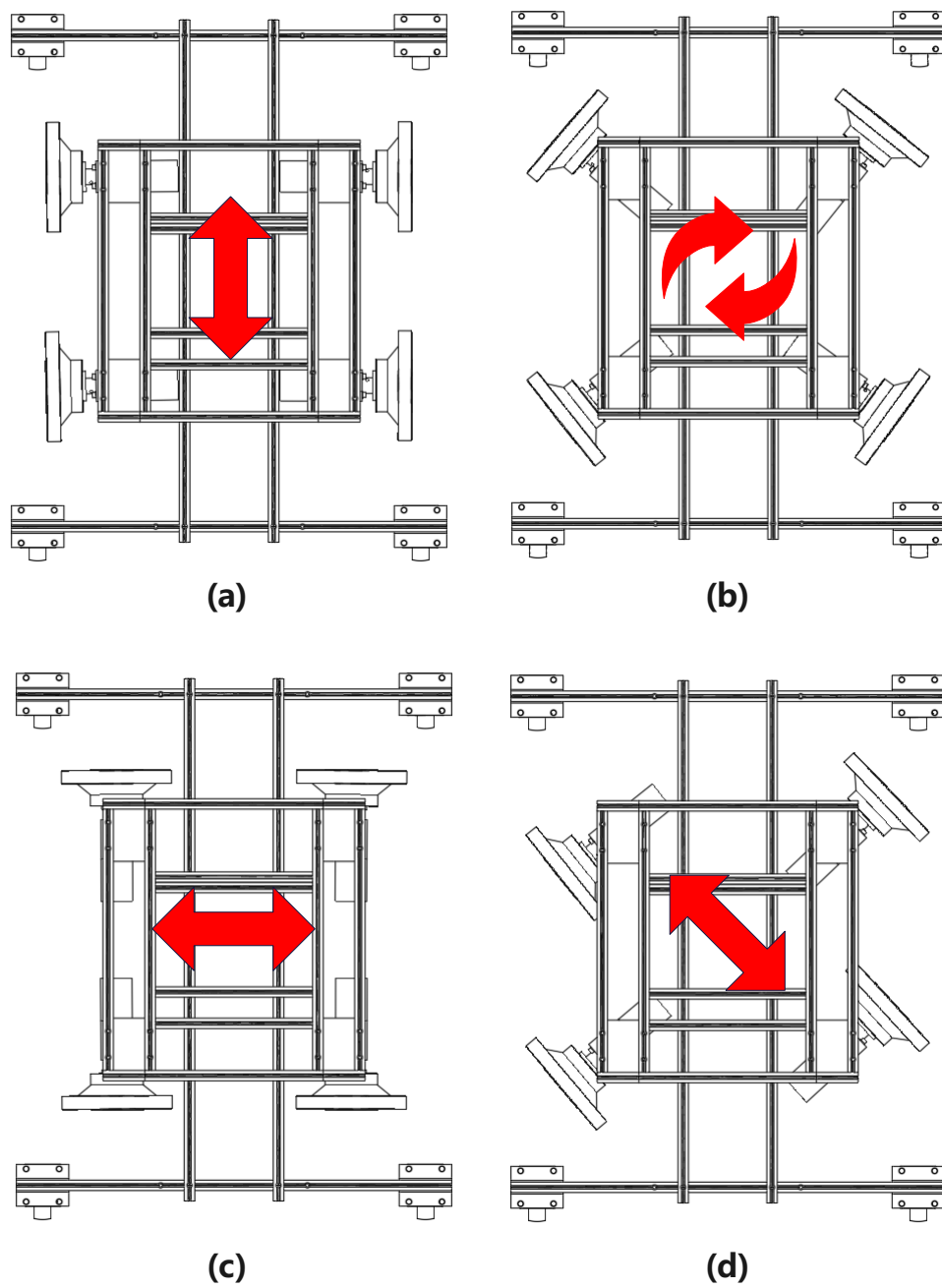


図 3.5: モバイルシャーシモジュールの移動方式の例. (a) 前進と後退 ; (b) 超信地旋回 ; (c) 横に移動 ; (d) 対角線に動く.

しいと予想されている。寿命を延ばすために補助サポートモジュールが必要になる。このモジュールは断面  $20 \times 60\text{mm}$  と  $20 \times 80\text{mm}$  のアルミフレーム、キャスト、アルミプレートとアルミアングルから構成されている。ロボットの重量を分担する機能を果たしている。なお、ホイールは、補助サポートモジュールを装着していても消耗が激しいため、一定な頻度で交換する必要がある。

本研究で実験を行った温室だけではなく、3.1 節で紹介したように、世界中でさまざまな地域の温室にはコンクリートの床と温水パイプが存在しているため、ロボットはコンクリートの床と温水パイプに走る必要がある。そのため、ホイールは特別なデザインに設計されている。図 3.6 に示すように、直径  $210\text{mm}$  のロードホイールはコンクリートの床の走行に使用され、直径  $100\text{mm}$  のレールホイールは温水パイプの走行に使用される。温水パイプは床に直接置かれるのではなく、サポートデバイスによって支えられている。そのため、ホイールは約  $130$  度の傾斜角度でできている。ロボットがこのデザインにより温水パイプとのクリアランスを維持しながら脱線せず、サポートデバイスによる障害をスムーズに通過できる。強度を確保するために、プリントする時のパラメータの壁の厚さは  $2.4\text{mm}$  に設定され、インフィル密度は  $20\%$  に設定されてプリントされる。

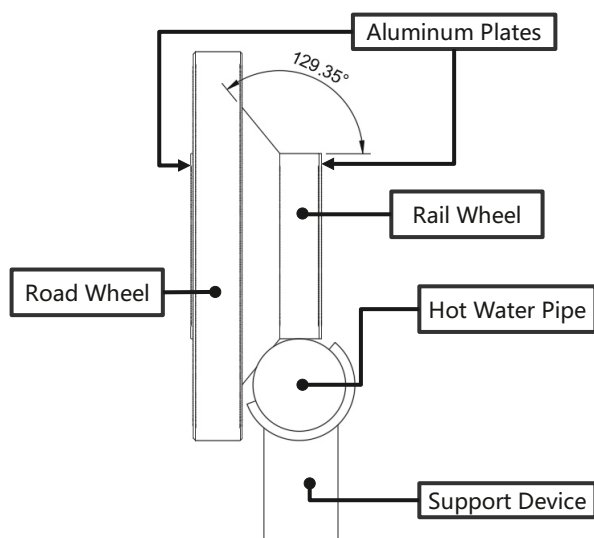


図 3.6: ホイールと温水パイプの断面図。

### ネスタブルモジュール

温室においてすべてのトマトの画像を撮ることは伸縮可能な機構が不可欠となる。本研究ではネスタブルで伸縮可能なデザインを提案している。この機構をネスタブルモジュールと命名し、直方体フレームモジュール (Cuboid Frame Module)、牽引モジュール (Traction Module) とカメラの昇降モジュール (Camera Lifting and Lowering Module) で構成されている (図 3.7)。

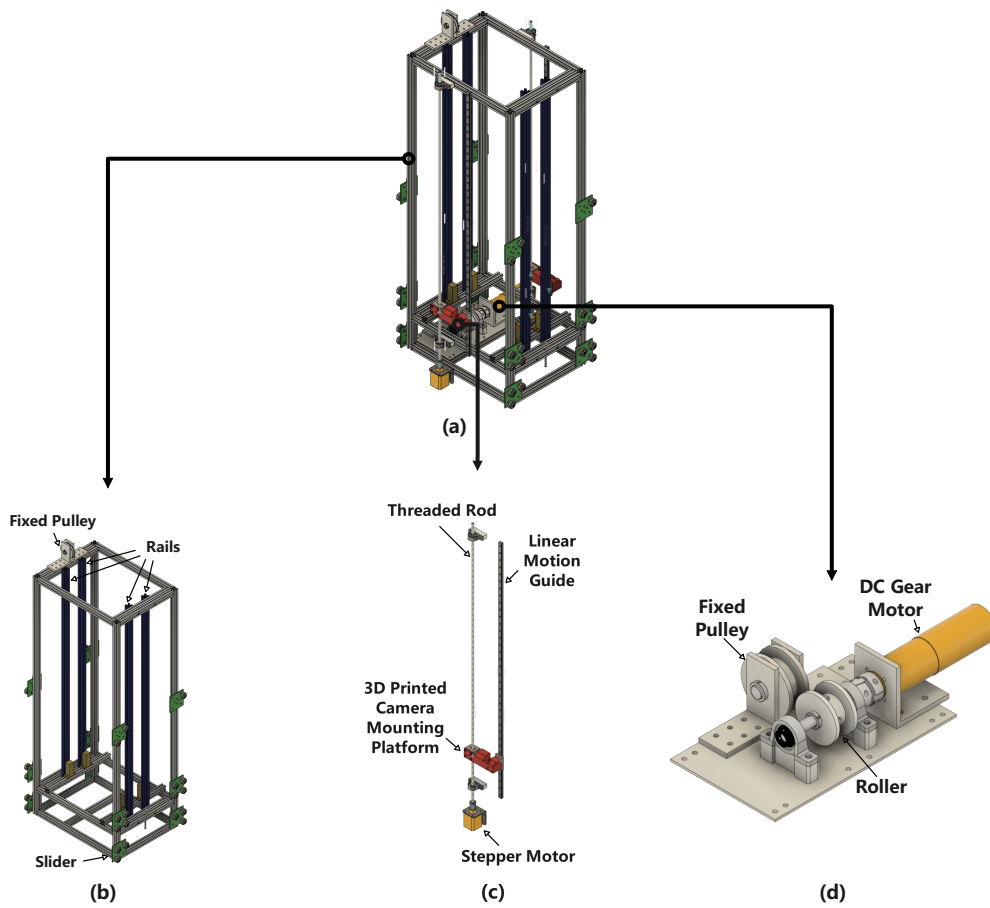


図 3.7: ネスタブルモジュールは3つのモジュールに分けることができる。(a) 直方体フレームモジュール；(b) カメラの昇降モジュール；(c) 牽引モジュール。



図 3.7(b) と図 3.8 に示すように、直方体フレームモジュールの形は断面 20×20mm のアルミフレームで作られている。外側には柱毎に3つ、総計 12 個のガントリープレートがインストールされ、内側には4つのガイドレールが取り付けられる。図 3.9 に示すように、赤い丸で囲んだところに示した通り、ガントリープレートはガイドレールにはめることができ、そしてガイドレールに沿って滑るようになっている。結果的に、小さいの直方体フレームモジュールは、それよりも大きい直方体フレームモジュールの中に入れてガイドレールに沿って上下に滑らせて動かすことができる。また、直方体フレームモジュールの上部に1つの定滑車が設置されている。

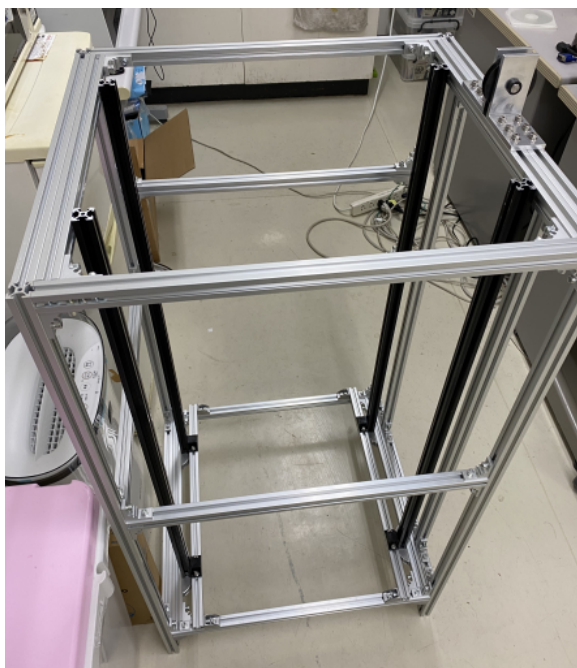


図 3.8: 直方体フレームモジュールの実物画像。

牽引モジュールは直流ギアモータ、定滑車とローラで構成されており、直方体フレームモジュールの最下部にインストールされている（図 3.7(d)）。直流ギアモータのトルクの伝達にワイヤを使用している。ワイヤの一端はローラに固定され、牽引モジュールに取り付けられた定滑車と直方体フレームモジュールの上部に取り付けられた定滑車を通過した後、もう一方の端は小さい方の直方体フレームモジュールに固定する。これで直流ギアモータがローラを駆動すると、ワイヤが小さい方の直方体フレームモジュールがガイドレールに沿って上昇したり降下したりすることができる。また、すべてのネスタブルモジュールに牽引モジュールがインストールされることはなく、他のネスタブルモジュールを持ち上げる必要があるかどうか次第となっている。明らかであるが、最上部のネスタブルモジュールには牽引モジュールをインストールする必要がない。また、ネスタブルモジュールがリミット位置に達したら、静止状態は直流ギアモータの逆トルクに依存する。

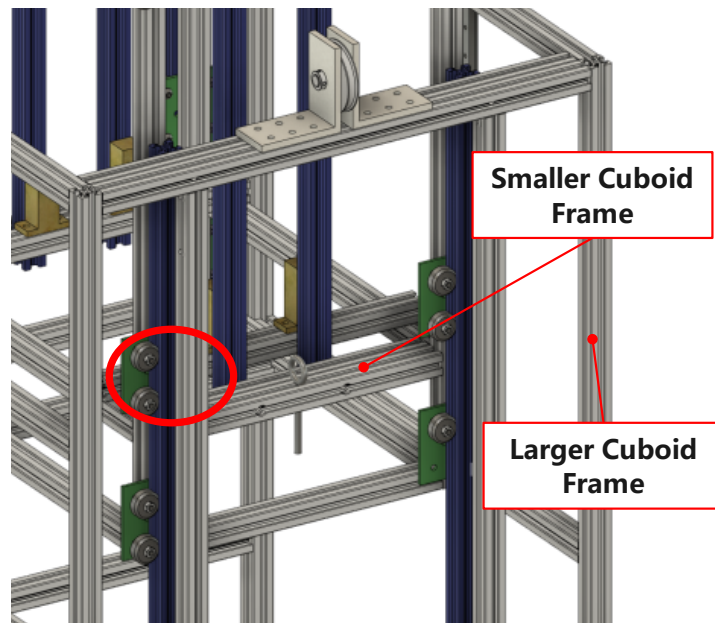


図 3.9: 小さい直方体フレームモジュールは大きい直方体フレームモジュールの中で上下に滑れる。

トマトの画像データを収集するために、直方体フレームモジュールの両側にカメラの昇降モジュールが取り付けられている（図 3.7(c)）。カメラの昇降モジュールはステッピングモータ、リニアモーションガイド、送りねじ棒、および、カメラマウントプラットフォームで構成されている。この中で、カメラマウントプラットフォームは取り外し可能である。なぜなら、トマトの実や葉などを撮るためにさまざまな角度から撮影する必要があり、それに応じてカメラを変更する必要がある可能性があるためである。ユーザーが具体的なタスクによって配置できるようにするため、カメラマウントプラットフォームは 3D プリンタで作成する。これにより、タスクに対してカメラの向きは変えられ、形状の異なるカメラにも柔軟に対応できる。ステッピングモータの回転により、カメラマウントプラットフォームがリニアモーションガイドに沿って上下に移動することが可能となる。よって、カメラマウントプラットフォームに搭載されているカメラは移動しながら画像データを収集することが可能になる。

## 配置

実際にロボットを使う時、モバイルシャーシモジュールは必須であるが、ネスタブルモジュールを何個使うのかはトマトがガイドされている高さに応じて決定する。

例えば、トマト植物の高さが 1m 未満の場合は、必要なモバイルシャーシモジュールの上に、ネスタブルモジュールを 1 つ使用する。トマトの茎が 2m までガイドされている場合は、ネスタブルモジュールを 2 つ使用する。トマトの茎が 3m までガイドされている場合はネスタブルモジュールを 3 つ使用する。図 3.10 に示すように、ネスタブルモジュールが伸展する

ことでモニタリングの範囲を拡張し、全てのトマトをカバーすることが可能になる。

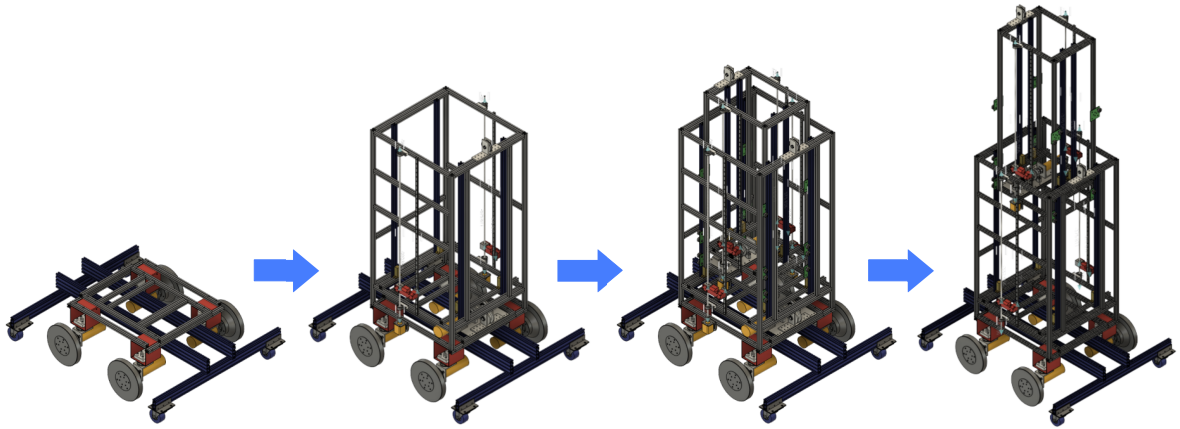


図 3.10: ロボットの配置の例.

当然のことながら、4つのネスタブルモジュールを使うと最大4mまでガイドされたトマトの植物に対応できる。ユーザは具体的な状況に対して決定することができる。本研究で実験を行った温室ではトマトの茎が約3mまでガイドされているため、図3.11に示すように、1つのモバイルシャーシモジュールと3つのネスタブルモジュールを用いてロボットを作成した。全体の重さは約70kgで、スタンダードモードの高さは約170cm、伸展モードは約300cmとなる。

実験を行った際には、ネスタブルモジュールの両側に1個ずつのRGBカメラを装着している。つまり、カメラの数はネスタブルモジュールの数の2倍となる。ネスタブルモジュールはそれぞれ独立しているため、4個のネスタブルモジュールを使う場合、総計8箇所のトマトを同時撮影することができる。大規模のトマト温室にしても効率が高いと考える。このように、モジュール毎に独立にデータ収集を行い、更にモジュールの中でも並行してデータ収集を行うことにより、作業効率が向上し、データ収集にかかる時間が大幅に短縮される。

### 3.3.2 ハードウェアシステム

ロボットのハードウェアは各モジュールにインストールされている。図3.12に示すように、各モータ及び直流モータドライバの他に、モバイルシャーシモジュールには、Jetson Xavier NX, Arduino Mega, ステレオカメラ, および USB ハブが配置されている。

Jetson Xavier NX はメインのコンピュータで、全ての計算を行い、信号を Arduino までに送る。4つの直流モータドライバは4つの直流ギアモータに接続し、Arduino Mega によって制御される。Arduino Mega はサーボモータも制御する。ステレオカメラは Jetson Xavier NX に繋ぎ、ロボットの位置と姿勢を推定するために使用され、ロボットの真正面に取り付けられている(図3.13)。USB ハブは他のモジュールへのインターフェイスとしての役割を担う。さ

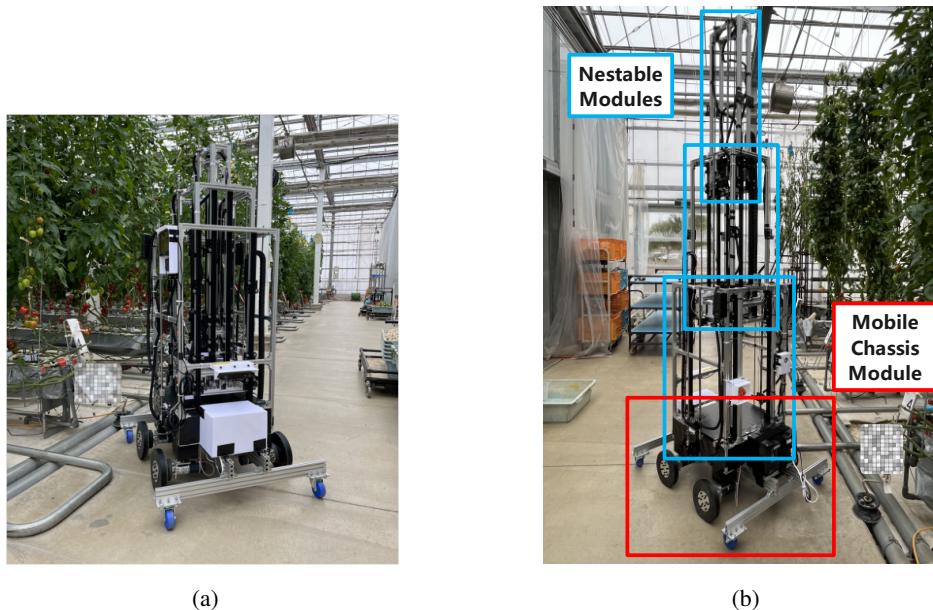


図 3.11: 本研究では作成したロボットの実物画像. (a) スタンダードモード ; (b) 伸展モード.

らに、ワイヤレスのジョイスティックが Jetson Xavier NX に接続されており、ロボットを手動制御するために使われる。

ネスタブルモジュールには Arduino Nano, 直流モータドライバ, 2つのステッピングモータドライバ, 2つのリミットスイッチ, 2台のカメラと USB ハブを搭載している。Arduino Nano はドライバを通して直流ギアモータとステッピングモータを制御する。図 3.14 の示すリミットスイッチには「1」と「0」2つの状態がある。これはロボットの伸縮機能を実現する最も重要なセンサであり、ネスタブルモジュールの上と下毎に1つずつがインストールされている。ネスタブルモジュールがもう1つのネスタブルモジュールの牽引モジュールにより上昇や降下する際に、金属のプレートに接触したらリミットスイッチの状態が変わる。そして Arduino Nano がその状態を読み取り、ネスタブルモジュールの上昇や降下を止める。この仕組みによりロボットの伸縮を実現している。

ネスタブルモジュールの中の Arduino Nano と2台のカメラが USB ハブに接続されており、モバイルシャーシモジュールの USB ハブに接続している。実験ではネスタブルモジュールを3つ作成したため、図 3.12 では3つのネスタブルモジュールを示しているが、実際はネスタブルモジュールの数は任意である。なお、最も小さいネスタブルモジュールには内側のネスタブルモジュールを持ち上げたり降ろしたりする必要がないため、リミットスイッチや DC モータドライバなどを省略している。

ロボットの稼働中の状態をユーザが把握するために、ヒューマンマシンインターフェイス (Human Machine Interface: HMI) が重要である。これにより、ユーザがロボットをよりうまく制御できるようになるだけでなく、ユーザの安全にも関わる。提案するロボットでは、グラフィカルユーザインターフェイス (Graphical User Interface: GUI) と音声再生で HMI を実

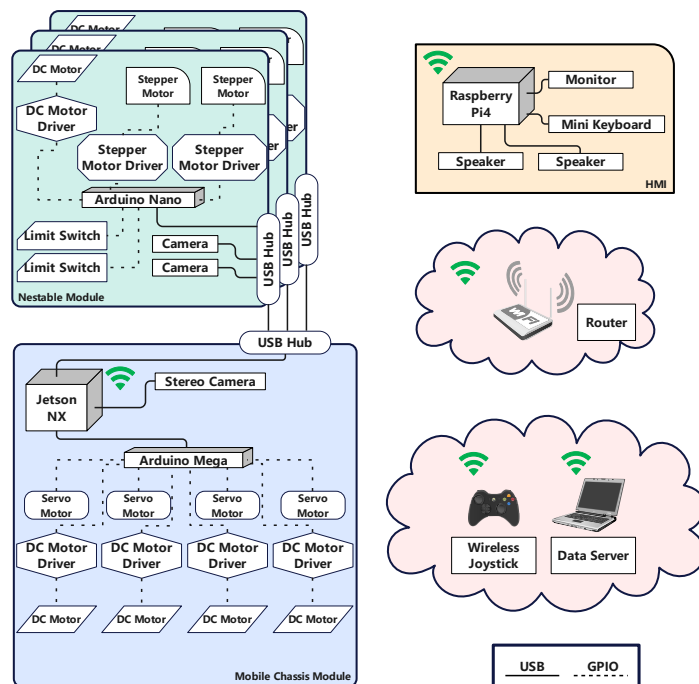


図 3.12: ハードウェアシステムのブロック図. 3つのネスタブルモジュールが使用されることを例として示されている.

現するため、Raspberry Pi4, モニタ, マウス一体型ミニキーボード, およびスピーカがインストールされている. 図 3.15 に示すように, ユーザは, マウス一体型ミニキーボードを使ってボタンを押し, 関連する機能は実行する前にスピーカからユーザに通知し, 本当に実行するかどうかはダブルチェックが行えるように設計されている. また, 実行中の状態に関してもスピーカから音声によりアナウンスする. この HMI を通して, ユーザがロボットを制御することができ, 稼働状態の認識も非常に容易になる. Raspberry Pi4 と Jetson Xavier NX はルータを介して通信する.

### 3.3.3 電気システム

ロボットの電源は定格 500W, 150Ah のポータブルバッテリーを用いている. 図 3.13 に示すように, バッテリーはモバイルシャーシモジュールの背面に直接固定され, 110V AC 電源コンセントを提供している. ポータブルバッテリーを使用する利点はいくつかある. まず, バッテリーの携帯性により, 充電が容易になる. バッテリーを取り外して充電できるため, 充電時にロボットをコンセントに近づける必要がない. もう 1つの利点はバッテリーをすばやく交換でき, モニタリングの範囲が広い場合には, より大容量のバッテリーに変更して使用できることである. これにより, 柔軟なモニタリングの計画を立てられる. さらに, 過熱に対する保護機能も備えているため, 自律稼働する時にロボットが発火する危険を避けられる.

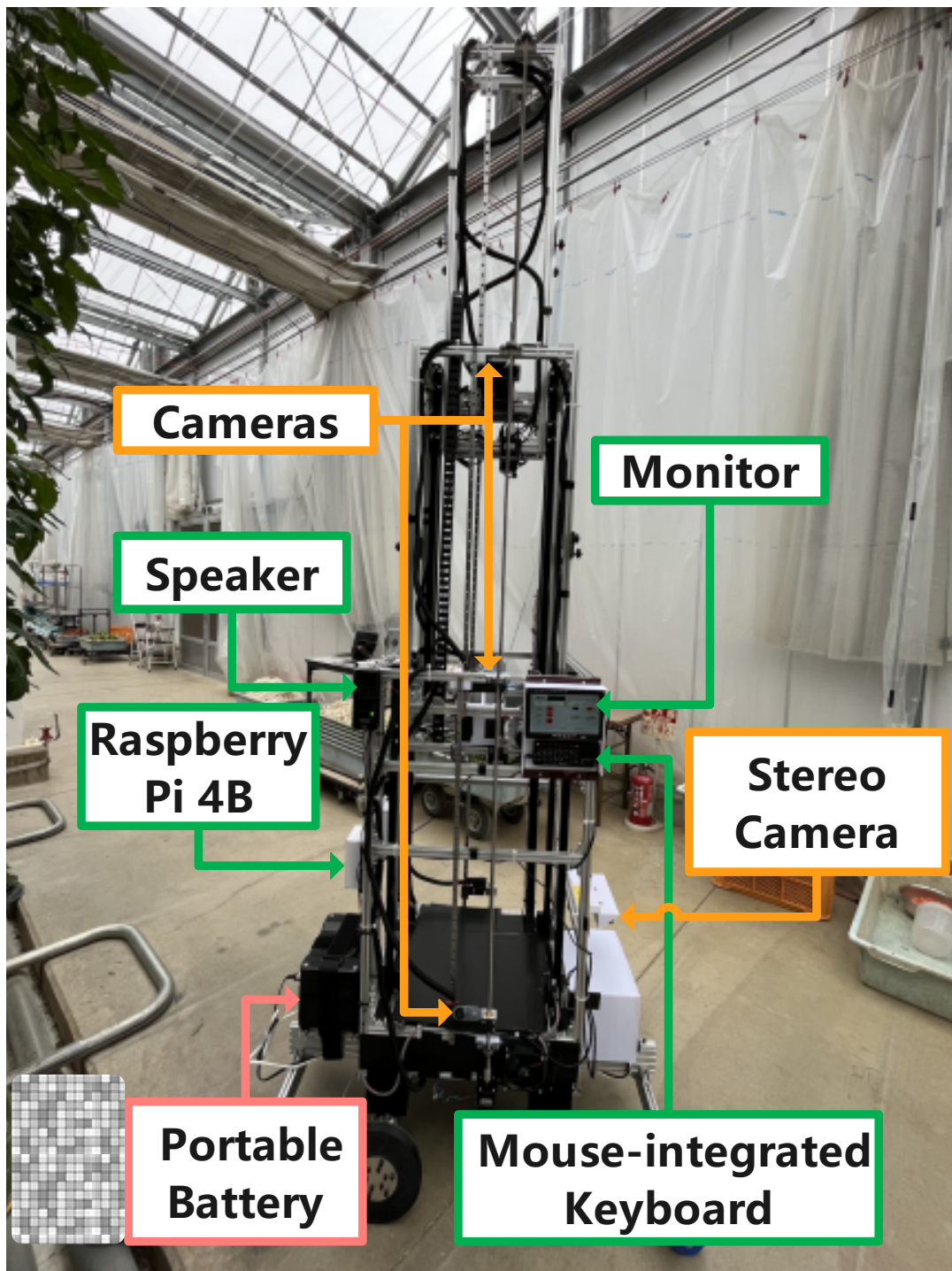


図 3.13: ロボットにインストールされているハードウェア.

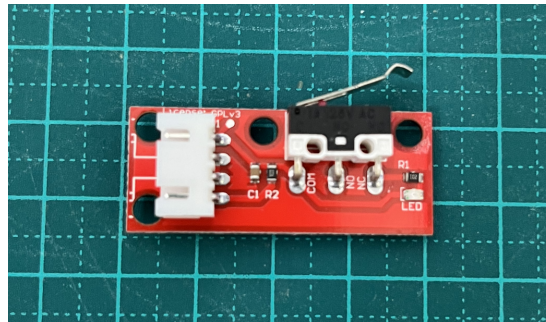


図 3.14: ロボットの伸縮状態を把握するリミットスイッチ.



図 3.15: グラフィカルユーザインターフェイスの例. ユーザはこのインターフェイスを通してロボットに指示する. ボタンは実験に対応している.

また、ポータブルバッテリーの110VAC電源を12Vの直流電源に変換するためには、AC-DCコンバータを用いる。この12Vは回路の入力電源になっており、直流ギアモータやサーボモータなどに提供している。回路の入力電源のON/OFFにはリレーが使われている(付録A.1, A.2)。メインのコンピュータはトランジスタアレイを通してリレーを制御している。同時に、非常ボタンもリレーを制御することができる。つまり、人かロボット自身か、どちらからでもロボットを非常停止することができる。このデザインにより安全性が高まる。また、リミットスイッチなどのセンサは5Vが必要のため、3端子レギュレータを用いて5Vの電源を作り出している。

ロボットは温水パイプの上で走行するため、モバイルシャーシモジュールの回りは周囲の環境よりもわずかに温度が高い。さらに、直射日光にあたるとさらに高くなる可能性がある。ロボットに搭載されているメインのコンピュータの発熱も考慮すると、モバイルシャーシモジュールの側面に冷却ファンを追加することは必須である。冷却ファンのON/OFFもリレーを通してメインのコンピュータにより制御されている。

### 3.3.4 ソフトウェアシステム

Jetson Xavier NX と Raspberry Pi4 は Ubuntu18.04 がインストールされている。ロボットのソフトウェアシステムはROSに基づいて開発されており、モジュール化されている。

図3.16で緑色で示されている「インターフェースモジュール」はRaspberry Pi4にインストールされている。インターフェースモジュールの役割は、GUIを表示し、ユーザの指示を「データ収集モジュール」と「手動コントロールモジュール」に送り、「データ収集モジュール」からのフィードバックを反映することである。さらに、「状態監視モジュール」からの情報でロボットの稼働状態も反映する。

図3.16でピンク色で示されている「デバイスコントロールモジュール」はArduino Nanoにインストールされているため、ネスタブルモジュールの数と関係がある。このモジュールは、「画像収集モジュール」から指示を受けて進捗状況のフィードバックを返す。また、最下部のネスタブルモジュールにインストールされているArduino Nanoには温度センサと電流センサをインストールされているため、センサから読み取ったデータを「状態監視モジュール」に送信する。

図3.16で黄色で示されている「画像収集モジュール」は「インターフェースモジュール」から開始コマンドを受け、温水パイプ上の自律走行と画像撮影を実行する。そして画像データ収集の実行結果を「インターフェースモジュール」に送り、ディスプレイに表示する。「状態監視モジュール」は「デバイスコントロールモジュール」から温度と電流のデータを受け、緊急ボタンの状態と電流の値などによって回路の入力電源のオンオフを制御し、温度の値によって冷却ファンのオンオフをコントロールする。回路の入力電源のオンオフ状況に基づいたデータは「インターフェースモジュール」に送られて表示する。「オドメトリモジュール」はステレオカメラに基づいて推定したロボットの位置と姿勢をパブリックする。オドメトリ情報に基づいてロボットは進行距離を推定することができる。「手動コントロールモジュール」は「インターフェースモジュール」から開始コマンドを受けたらアクティブになる。そ



して具体的な移動指示を青い「行動モジュール」に送る。「行動モジュール」は Arduino Mega にインストールされ、モバイルシャーシモジュールのモータの動きを制御する。

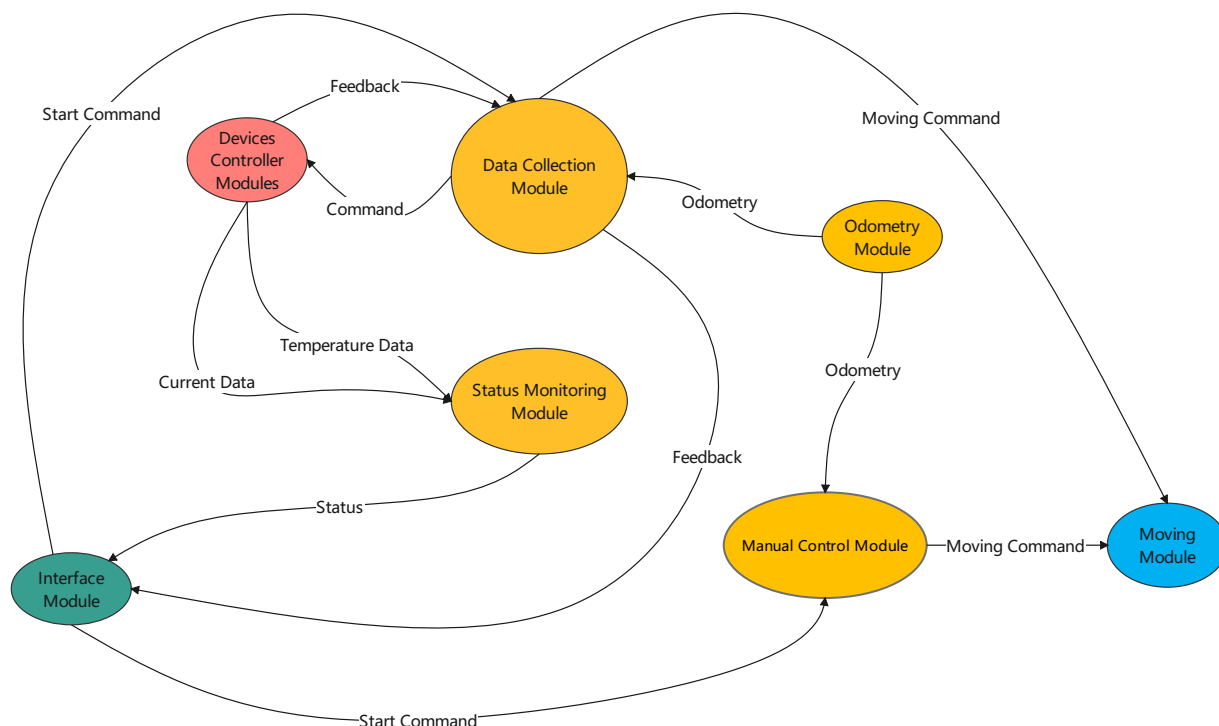


図 3.16: ソフトウェアシステムの構成. 異なる色はそれぞれのデバイスにインストールされていることを意味する.

### 3.4 データサーバ

データサーバは温室内のオフィスなどの場所に設置し、ローカルエリアネットワークを通してロボットと通信する。データサーバの役割は、ロボットの収集した画像データを保存することと病気検出モデルの含まれるプログラムを実行することである。ディープラーニングネットワークが含まれるプログラムを実行する必要があるため、GeForce GTX 1080 グラフィックカードを配置している。このグラフィックカードは2016年に販売が始まった古いタイプであるが、8GBのメモリを保有しながら、価格が低いというメリットがある。大型なネットワークをトレーニングすることはできないが、YOLOなどを実行することはできる。農業の場合、低コストであることは将来的に提案のシステムを普及させる際には重要な要素の1つである。

### 3.4.1 2段病気検出モデル

どんなネットワークや検出器であっても、必ず誤検出が出る。本研究で提案するシステムはユーザに病気のトマトの位置を報告するとユーザが病気のトマトに対して処置を行うことを想定している。そのため、検出器が病気でないトマトを病気であると検出する誤検出率が高すぎると、ユーザが処置する効率が悪くなる。一方で、病気のトマトを見落とす未検出率が高くなると収量の低下につながる。このことから、未検出を一定以下に抑えつつ、誤検出率をできるだけ低くすることが重要であることがわかる。

病気のトマトを検出するために、既存の画像分類用のディープラーニングネットワークを転移学習して用いることを考える。現在までに、非常に多くのネットワークが提案されているが、それらのネットワークの学習には同じデータが用いられている場合にも、ネットワーク構造はそれぞれ異なるため、画像から抽出される特徴もネットワークにより異なることが予想される。そこで、本研究では、複数のネットワークを導入して共通集合をとれば誤検出率が下がることが予想される。

ロボットにより収集される画像は複数のトマトの木になっている複数のトマトの実であるが、ユーザが知りたい情報は病気のトマトがどの実であるかという情報であることから、病気のトマトの領域検出は必須である。また、病気のトマトの画像かどうかの分類を行う場合、複数のトマトが含まれる画像よりもトマトの単一の果実を分類した方が精度が向上することが予想される。本研究では、まずできるだけ見落としのないように画像全体から物体検出ネットワークを用いて単一の病気のトマトの領域の候補を検出してから、その画像が病気のトマトかどうかを検証ネットワークで分類するという方針とする。

図 3.17 に示すように、本研究では性能のバランスを配慮した上で、病気検出モデルを2段構造にデザインしている。2段病気検出モデルは検出ネットワークと検証ネットワークで構成されている。

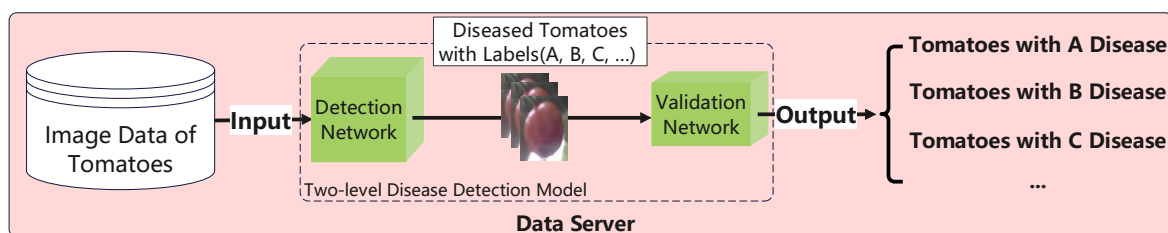


図 3.17: トマトの病気検出モデルは2段構造にデザインされている。

検出ネットワークは、画像から病気のトマトの果実が写っている領域を検出するネットワークである。検出結果は、トマトの病気のラベルと病気のトマトを囲むバウンディングボックスである。バウンディングボックスの座標に基づいて、病気のトマトが写っている領域が画像全体から切り取られる。検出ネットワークの結果だけで十分な検出精度を達成するのは難しい。そのため、検出ネットワークでは信頼度の低い領域も抽出し、切り取られた画像の領域全てを候補として検証ネットワークに入力し、さらに検証することにより誤検出を減らす。

検証ネットワークは、単一のトマトが写っている画像を病気のトマトかどうか分類するネットワークである。一般的に、画像からの領域検出に比べ、画像分類のネットワークは高い精度を達成している。そのため、検出ネットワークで抽出した領域が病気のトマトかどうかを検証するネットワークとして適切だと考えられる。検証ネットワークの出力は、検出したい病気に設定される。検証ネットワークの分類結果が検出ネットワークによって予測されたラベルと一致する場合には予測された結果は正解であると見なされ、一致しない場合には切り取られた画像はユーザのより目視で確認される目視検査エリアに移動される。

具体的な例を挙げる。検出ネットワークが1枚の画像から病気のトマトを1つだけ検出し、予測結果が「A」病であるとする。病気のトマトの切り取りは予測のバウンディングボックスの座標に基づいて取得され、検証ネットワークに入力される。検証ネットワークの予測も「A」病である場合、病気のトマトは「A」病と見なされる。そしてシステムは元の画像の位置情報に基づいてA病にかかっているトマトの位置を出力する。一方、検証ネットワークの分類結果は「B」病の場合、実際の結果は確定できないため、病んだトマトの切り取り画像は目視で確認する必要があるために目視検査エリアに配置される。この2段階病気検出モデルは不要な作業時間を避けるために設計されている。

### 3.5 システムのワークフロー

本論文を執筆する時点で提案のシステムは全自動ではなく、ロボットを温水パイプの頭部まで移動させる必要がある。また、簡潔に説明するため、本節の内容は温水パイプがあることを前提として述べる。温水パイプがインストールされていない温室の場合、以下の内容にある「温水パイプ」を「2列のトマト植物の間の通路」に差し替えれば良いと考える。

システムのワークフローは図3.18に示す。まず、ユーザがGUIを通して「手動モード」ボタンを選び、ジョイスティックを用いて手動でロボットをコントロールし、温水パイプの起点まで移動させる。誰でも簡単にロボットの制御を把握できるように、シンプルなキーの組合せで操作できるように設定されている。表3.1の示すように、ジョイスティックにある2個のキーを同時に押せば車輪の向き（移動パターン）を決められる。実際、実験を行った温室で操作を行った協力者は1時間前後でロボットの操縦法を把握することが可能であった。

表 3.1: キーの組み合わせでロボットを制御する。

キーの組み合わせ	動作
LB + 左スティック（上下に押す）	前進, 後退
LB + 右スティック（左右に押す）	後輪の方向転換
RB + 右スティック（左右に押す）	左右に旋回
LB + 左スティック（上下に押す） + 右スティック（左右に押す）	前進や後退しながら回る

その後、ユーザは「自動収集モード」ボタンを選択すれば、ロボットは自動的にスタンダードモードから伸展モードに変形し、温水パイプに沿って自律走行し、トマトの画像データを

収集する。収集が完了すると、ロボットはスタンダードモードに戻り、温水パイプの起点に戻る。ロボットは起点で収集した画像データをデータサーバに送信してからスタンバイモードに入り、ユーザからの指示を待機する。「自動収集モード」ボタンを選んでから、スタンバイモードに入るまでに全ての動作は自動的に行う。

ユーザの計画に従って、ロボットは次の温水パイプに移動させて上記のプロセスを繰り返すか、終了することができる。全ての収集タスクが完了したら、データサーバで実行されている病気検出のプログラムが画像データから病気のトマトを自動的に検出して具体的な位置をユーザに通知する。

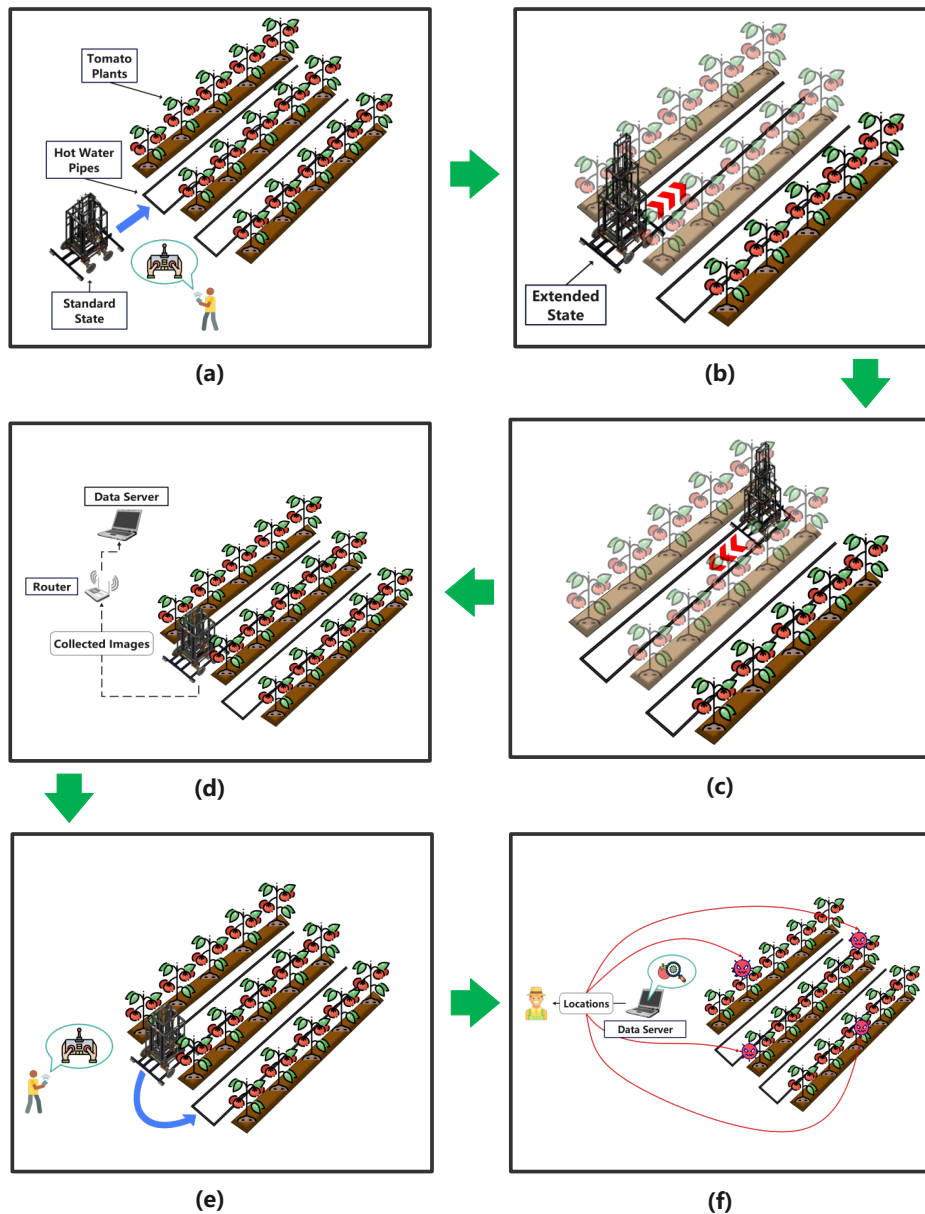


図 3.18: トマトの病気モニタリングシステムのワークフロー. (a) ユーザーがジョイスティックを用いてリモートでロボットを制御し、温水パイプの起点まで移動させる; (b) 「自動収集モード」が選択されたら、ロボットが伸展モードに変形し、温水パイプに沿って自律走行してトマトの画像データを収集する. (c) ロボットが事前に設置した位置に到達すると、スタンダード状態に戻り、そして起点に戻る. (d) 収集した画像データがサーバーに送信されてから、ロボットはそのままユーザの指示を待つ. (e) ユーザがロボットをコントロールして次の温水パイプに移動し、(b)(c)(d)が繰り返れる. (f) 最後はデータサーバで実行されている病気検出プログラムが病気のトマトの具体的な位置をユーザに通知する.

## 第4章 実験，結果とディスカッション

### 4.1 実験環境の紹介とロボットの配置

本研究の実験は共同研究先が保有する温室で行った。図4.1に示すように、本研究の実験を行った温室にも温水パイプがインストールされている。温水パイプは温度を制御するだけでなく、収穫用の台車や高所作業台などのレールとしても使われているため、本研究で開発した新型ロボットも温水パイプの上に稼働する。トマトは列に沿って栽培されており、茎は最大約3mまでガイドされている。



図 4.1: 実験を行った温室環境。

モニタリングの対象は約3mまで伸びているため、実験に用いたロボットは、モバイルシャーシモジュールと3つのネスタブルモジュールから構成されている。

### 4.2 モニタリングのための定義とセッティング

本システムは最終的にユーザに病気のトマトの具体的な位置を通知するため、画像データに位置情報を付ける必要がある。また、今回の実験では、モニタリングの対象は1列のトマトのみである。つまり、毎回画像データを収集する時にロボットの片側の3つのカメラのみ

を動かせば良い。従って、実験を行うため、モニタリングタスクや温室環境に合わせてセッティングと定義を行った。

## 定義

実験を行う温室の概略図を図4.2に示す。図4.2のように、モニタリングの対象となるトマトの植物の両側に温水パイプがあり、温水パイプに垂直な軸を  $X$  軸、パイプに平行な軸を  $Y$  軸と定義した。  $Y$  の最大値は 24 に設定され、2つの値の間の距離は 750mm に設定した。つまり、ロボットは温水パイプの上に総計 18 m の距離を進み、25 箇所のトマトの画像データを収集するものとした。このように設定することで、ロボットは全てのターゲットのトマトをカバーできるようになった。また、最下部のネスタブルモジュールは「L1」という名前を付け、真ん中のを「L2」とし、最上部のを「L3」と定義した。

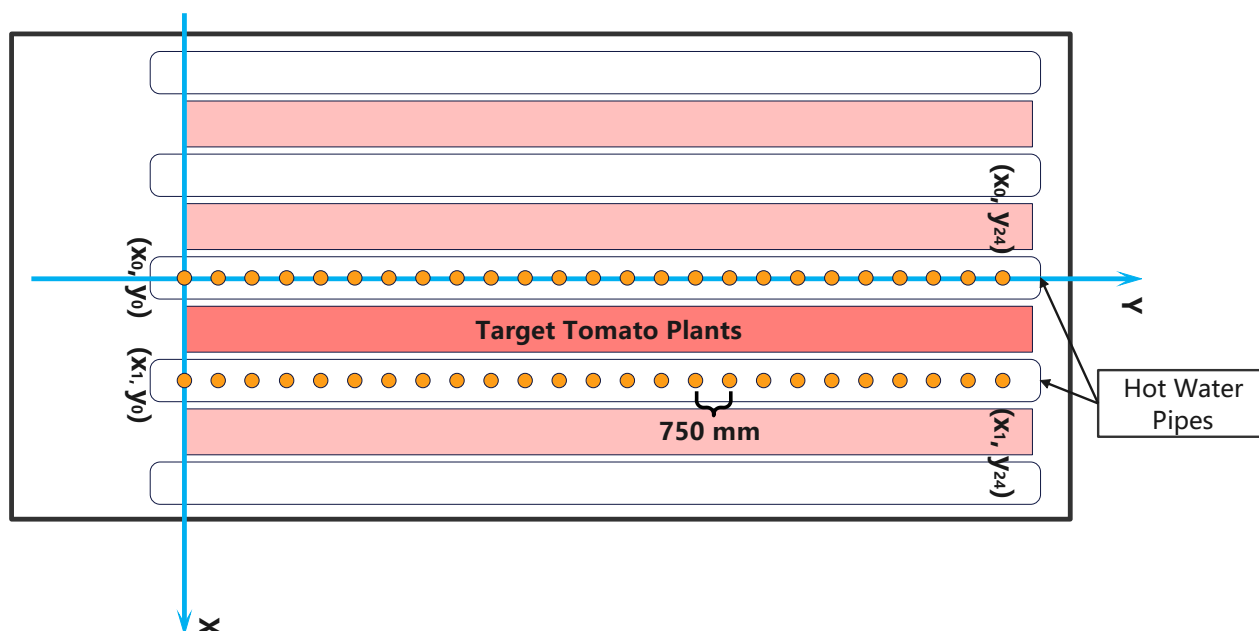


図 4.2: 実験を行う温室の概略図。ロボットはオレンジ色のポイントで自動停止してトマトの画像データを収集するように設定。

## セッティング

ロボットは温水パイプの上で自動的に稼働し、画像データを収集する具体的なセッティングを以下に示す。ロボットが  $(x_0, y_0)$  列で稼働する場合、まずは  $(x_0, y_0)$  で自動的に伸展する。

図 4.3 の上部に示すように、ロボットはまず L2 を上げて、限界に到着したら L3 を上げ始める。L3 が限界位置につくと伸展プロセスが完了する。スタンダードモードから伸展モードに変形するには 1 分 58 秒が必要であった。なお、この時間は直流ギアモータの出力に依存する。

伸展した後、図 4.3 の下部の赤い矢印が指しているように、ロボットの右側の 3 つのカメラ

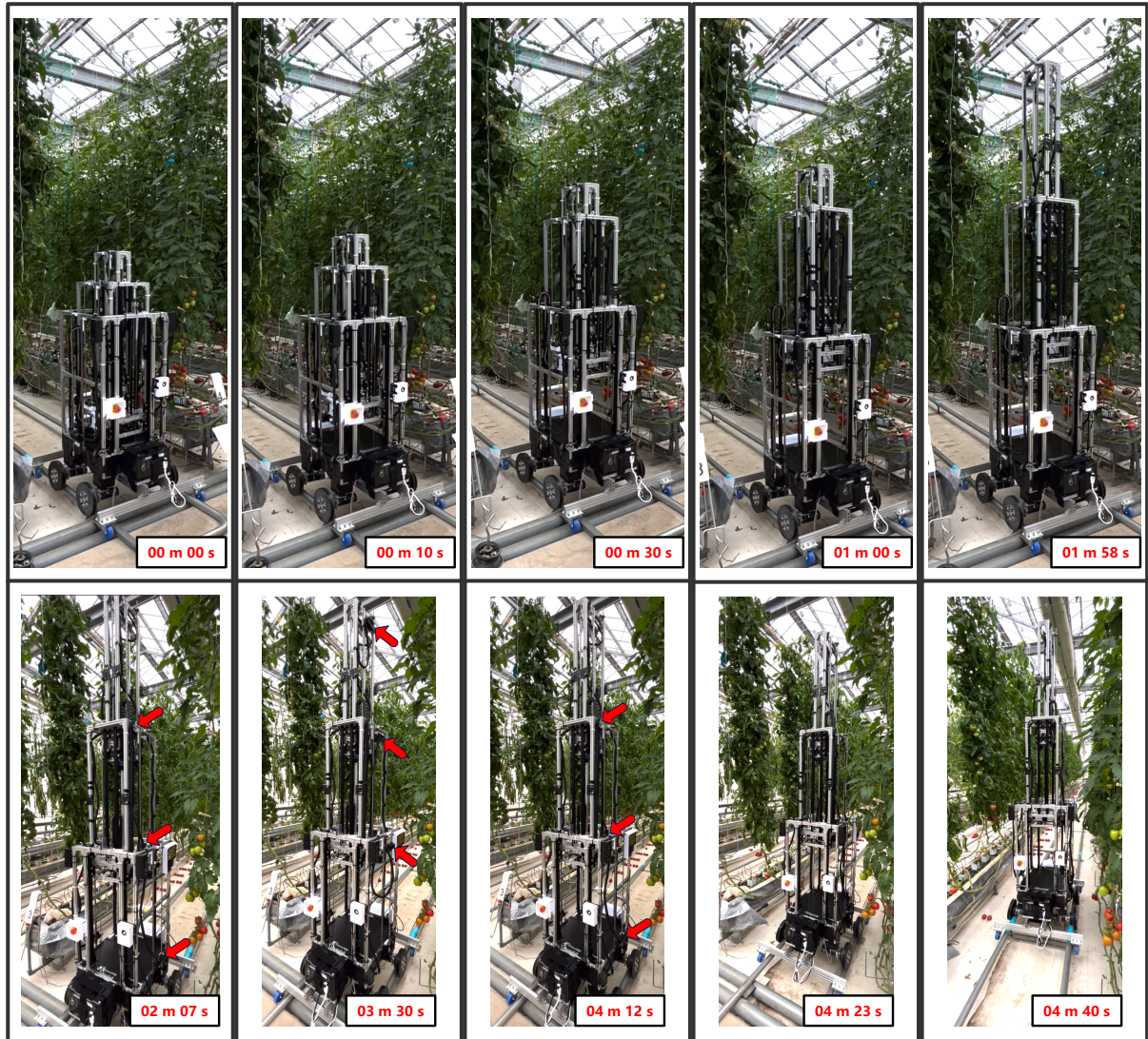


図 4.3: (上) スタンダードモードから伸展モードに変形するプロセス; (下) 3つのカメラ (赤い矢印が指しているところ) は同期に上昇し, 上昇した毎に3枚の画像データが収集される. 9回上昇したら元のポジションに戻り, ロボットは次の位置に移動する.



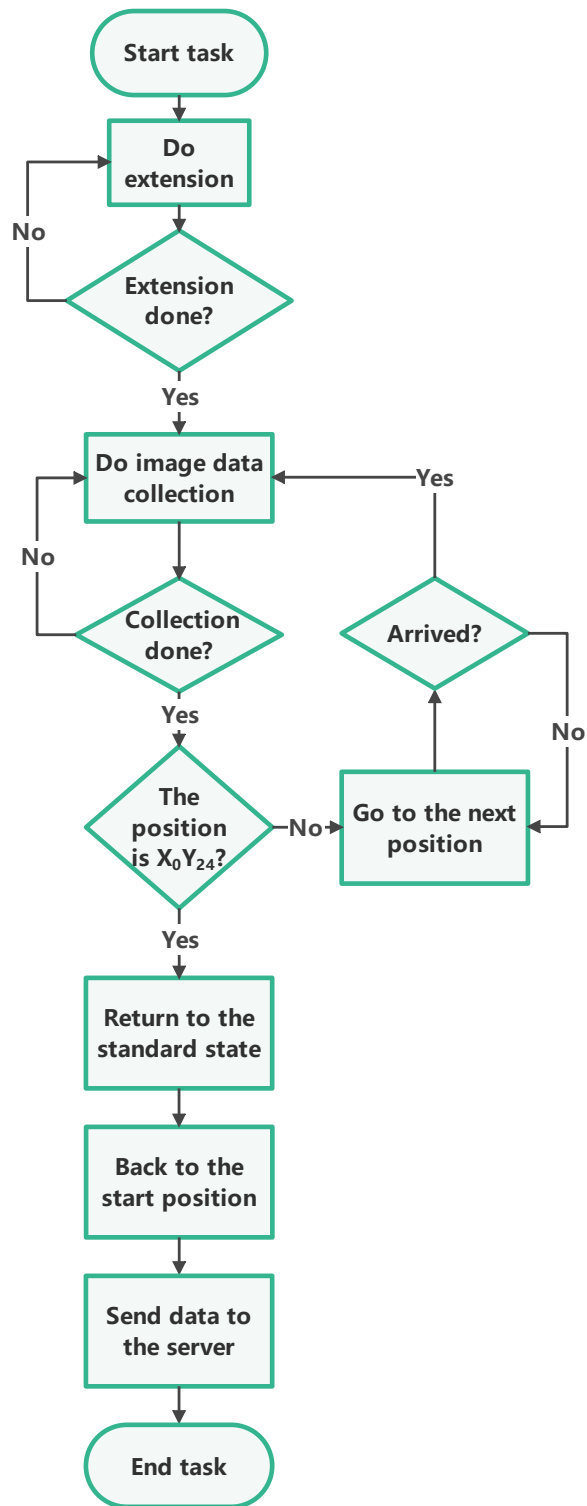


図 4.4: 温水パイプの上に画像データを収集するフローチャート.

は同時に上昇し、各上昇の間隔は 80mm に設定された。総上昇の回数は 9 回で、上昇するたびに各モジュールで 1 枚ずつ、合計 3 枚の画像を撮る。したがって、初期位置で撮った 3 枚の画像を加えて、 $(x_0, y_0)$  で 30 枚のトマトの画像を取得する。

次に、カメラは元のポジションに戻り、ロボットは自動的に  $(x_0, y_1)$  まで 750mm 進み、上記のプロセスを繰り返す。ロボットは  $(x_0, y_{24})$  で画像を収集し終わったら、スタンダード状態に戻り、自動的に  $(x_0, y_0)$  に戻る。最後に、ロボットは収集した画像データをデータサーバに送信し、待機状態でユーザの指示を待つように設定している。上記の流れは図 4.4 にまとめている。 $(x_1, y_0)$  列で稼働する場合の違いは、ロボットが左側のカメラを上昇して画像データ収集することである。

撮った画像データの名前は、日付、ロボットの位置、ネスタブルモジュールの番号、カメラのポジションで構成されている。フォーマットとしては「日付\_ロボットの位置\_ネスタブルモジュールの番号\_カメラのポジション」という形に設定された。例を挙げると、画像の名前が「20220401\_X0Y0\_L1\_0」となっている場合、まず、この画像は 2022 年 4 月 1 日に撮ったものになる。そして「X0Y0」とはこの画像はロボットが  $(x_0, y_0)$  で撮った画像であることを意味する。「L1」は最下部のネスタブルモジュールに装着されているカメラで、「0」は初期ポジションで撮った画像を意味する。言い換えると、「X0Y0\_L1\_0」に基づいて、システムは画像から検出した病気のトマトの位置を把握し、ユーザに「 $(x_0, y_0)$  列の頭で、右側の最下部にいくつかの病気のトマトがある」ように通知することができる。

また、 $y$  値がどこに対応するのかをユーザに明確にするために、黒いテープを温水パイプに貼りマーカとして用いる。

## 4.3 2 段病気検出モデルのトレーニング

ロボットが収集したトマトの画像データに基づいて、2 段病気検出モデルのトレーニング実験を行った。本節では、データセットの分割方法を変更して、複数のネットワークを異なるデータセットでトレーニングし、結果によって最も性能のよい物体検出ネットワークと分類ネットワークを選択した。そして、2 段病気検出モデルでの検出結果と物体検出ネットワークのみでの検出結果を比較した。

### 4.3.1 データセット

本研究では、実験を行った温室で発病しやすいトマトの尻腐れ果のみに注目した。文献 [56] によれば、尻腐れ果はトマトの重要な病気の 1 つであり、最初は果実の先端部分に水浸状の症状が現れる。そして水浸状部分は拡大していき、黒い斑点が現れる。だんだん黒い斑点も拡大し、先端部分が黒くなる (図 4.5)。尻腐れ果は伝染しないが、環境要因で引き起こされるため、タイムリに処置しないと、他のトマトにも発生する。

尻腐れ果を検出する適切な 2 段病気検出モデルを作るため、検出ネットワークと検証ネットワークを学習させ、よい性能のものを選ぶ必要がある。そして、学習するにはデータセッ

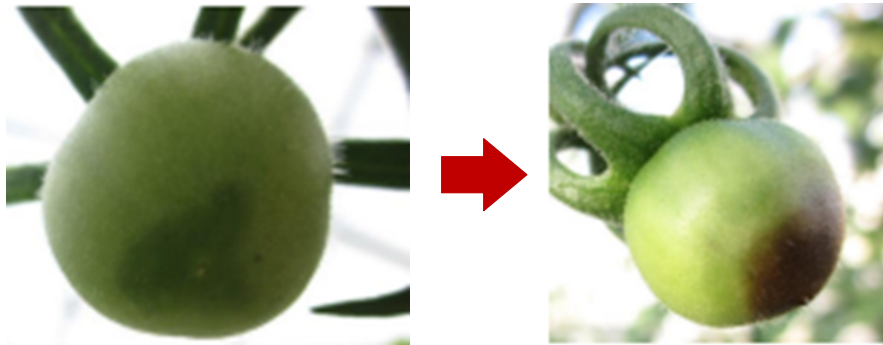


図 4.5: 尻腐れ果の症状遷移 [56].

トが必要となる。

インターネット上には、公開されているトマトのデータセットが多数あるが（例として [57-59]）、本研究では使用できない。理由は2つある。1つ目は、実験を行った温室で栽培されている実験の品種が新しく、通常のトマトとは形が異なることである。2つ目は、本論文の実験は尻腐れ果に焦点を当てており、トマトの正面または上から撮影した画像ではトマトの先端部分が見えないが、通常のデータセットに含まれるトマトの画像のほとんどが正面または上から撮影された画像であるためである。これらの理由から、ロボットによって収集された画像データのみを本研究のデータセットとして使用する。

4.2 節に示したセッティングに従って、ロボットは2022年4月4日から6月20日までの29日間温室内で稼働させた。ロボットの両側のカメラはトマトの先端部分の画像を取得するために45度上向きに設置した。ある場所で収集された画像データを図4.6に示す。画像を1枚撮影する度にカメラが1回上昇するため、画像の連続性が見られる。似ているところもあるが、それぞれ違う角度から撮ったもので、内容は異なっている。

収集したRGB画像のサイズは640×480であった。2.4.2項で紹介したとおり、Makesense.aiを使用して画像に含まれる尻腐れ果にかかっている全てのトマトに対し、サイズに関わらず、また、一部オクルージョンで隠れているものも含めて、ラベルを付けた。代表的な例を図4.7に示す。

次に、画像サイズを640×640にリサイズし、ネットワークのトレーニングのためにYOLOおよびCOCOフォーマットで保存した。近い日付の画像はほとんど変化がないため、一定な間隔を持つ画像にアノテーション作業を行った。実験を行った時点で、4月4日、4月13日、4月21日、5月1日、5月9日、5月16日、6月7日、6月13日、6月20日の画像データにラベルが付けられた。そして、尻腐れ果にかかっているトマトが含まれる画像は合計2559枚となり、これを用いて学習や評価のためのデータセットとして使用した。

従来のトレーニング方法では、データセットを、トレーニングセット、バリデーションセット、およびテストセットに分割する。ネットワークはトレーニングセットを使用してトレーニングし、バリデーションセットを使用してトレーニング中の性能を観察し、テストセットを使用して学習済みのネットワークのパフォーマンスを確認する。本研究のデータセットは、

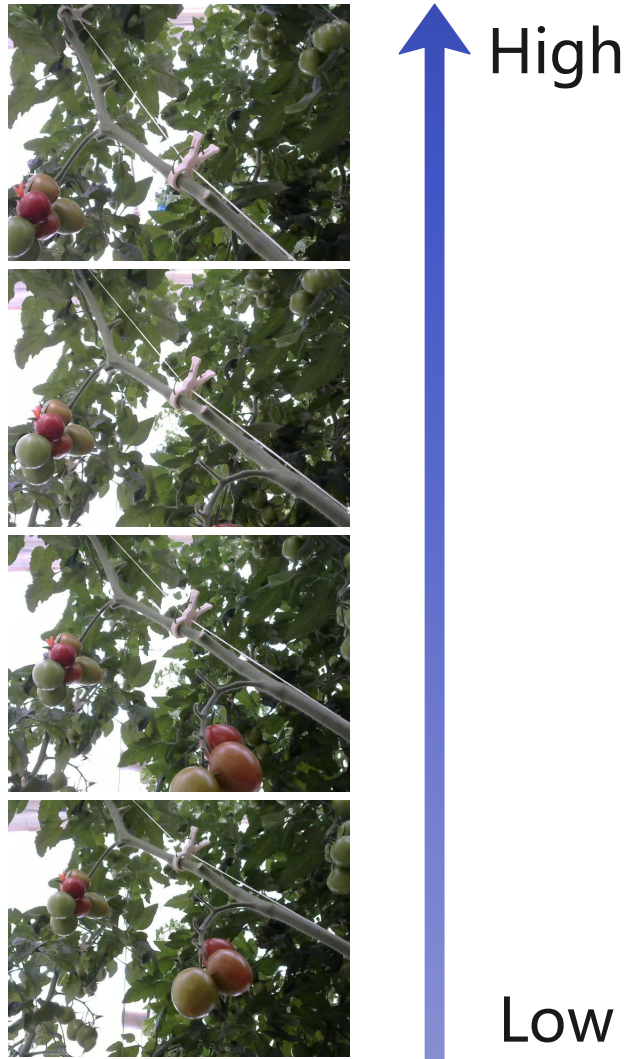


図 4.6: ある場所の画像データの一部. 画像の撮る順序は現実世界の低処 (Low) と高処 (High) に対応する.



図 4.7: ラベル付けされたトマトの例.

通常学習に用いられるデータセットとは異なり，時間と場所にある程度関係したデータが含まれている．そのため，最もよいデータセットの分割方法を確認するために，データセットを3つの条件で分割した．

**データセット1** 一定な割合でランダムに分割．データセットを6:2:2の割合でランダムにトレーニングセット，バリデーションセット，テストセットに分割した．

**データセット2** 日付による分割．4月4日，4月13日，5月1日，5月9日，6月7日，6月13日の画像データをトレーニングセットとして使用し，5月16日のデータがをバリデーションセットとして使用し，4月21日と6月20日のデータをテストセットとして分割した．

**データセット3** 場所による分割．図4.8に示すように，青いボックスで示した場所で得られたデータはトレーニングセットとして，赤いボックスで示した場所で得られたデータはバリデーションセットとして，緑のボックスで示した場所で得られたデータはテストセットとして分割した．なお，図中の各場所には，2箇所撮影した画像データが対応している．たとえば，「0」には $(x_0, y_0)$ と $(x_1, y_0)$ で収集された画像が含まれる．

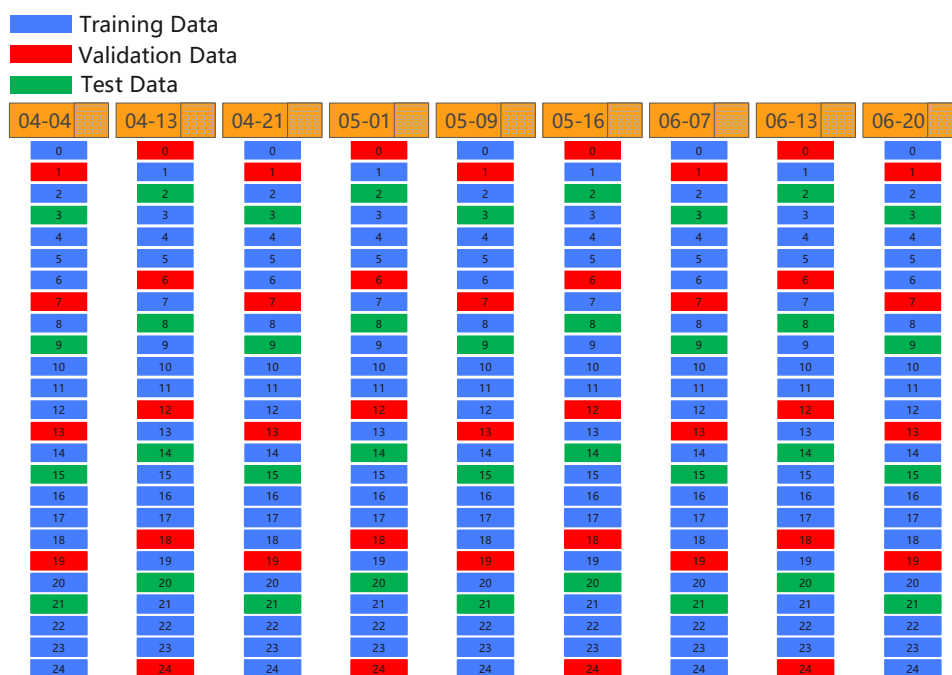


図4.8: データセット3の分割模式図．場所に基づいて分割されており，トレーニング，バリデーションおよびテストに用いるデータの撮影位置が異なっている．

3つの分割条件で作成されたデータセットに関する情報は表4.1に示す．いずれのデータセットにも，トレーニングセット，バリデーションセット，およびテストセットと同じ比率

でそれぞれ総計 30 枚背景のみの画像を追加した。データセット 1 では、トレーニングセットに 1551 枚の画像があり、バリデーションセットに 520 枚の画像があり、テストセットに 518 枚の画像があった。データセット 2 では、トレーニングセット、バリデーションセット、およびテストセットに含まれる画像の数はそれぞれ 1776 枚、436 枚、および 377 枚であった。データセット 3 のトレーニングセットには 1692 枚、バリデーションセットには 447 枚、テストセットには 450 枚の画像が含まれていた。

表 4.1: 3つの分割条件で作成したデータセットの詳細。

	Training Set	Validation Set	Test Set	Total
Dataset1	1551	520	518	2589
Dataset2	1776	436	377	2589
Dataset3	1692	447	450	2589

### 4.3.2 検出ネットワークのトレーニングと結果

提案のシステムを農業に普及するには、ロボットだけでなくサーバも可能な限りコストパフォーマンスを高くしなければならない。しかし大規模なネットワークの高速な推論には高性能のグラフィックスカードが必要であり、コストが高くなる可能性がある。したがって、物体検出ネットワークのサイズとパフォーマンスのバランスを考慮して、実験では、YOLOv5, YOLOv7 [60], Faster R-CNN [61] の一部のバージョン、および RetinaNet [62] を選択してトレーニングした。すべてのネットワークは最終出力を 1 つのクラスに変更した。つまり、尻腐れ果にかかっているトマトのみを検出するようにトレーニングした。他のすべてのトレーニングパラメータは公式の推奨するデフォルトパラメータを採用した。

作成した 3 つのデータセットをトレーニングに用いた。本論文では実用的かつ直感的に意味のある 3 つの指標を用いる。まず、mAP@0.5 はネットワークのパフォーマンスの重要な指標であり、値が高いほどパフォーマンスが優れていることを示す。Recall と Omission は正反対の指標であり、Recall は画像から検出された正しいオブジェクトの割合を反映し、Omission は偽陰性の割合を表す。実験の尻腐れ果の場合、Recall は尻腐れ果にかかっているトマトの検出率を反映し、Omission は見逃し率を反映する。デフォルトでは、ネットワークは mAP@0.5 と Recall の値を出すため、Omission は手動で計算する。式は次のとおりとなる。

$$Omission = 1 - Recall \quad (4.1)$$

トレーニング結果は表 4.2, 表 4.3, および表 4.4 に表示する。その中で、YOLOv5l の各指標はデータセット 1 でトレーニングしたときに最高であった。要するに、YOLOv5l は一定な割合でランダムに分割したデータセット 1 でトレーニングすれば最高のパフォーマンスを達成することができる。したがって、YOLOv5l は 2 段病気検出モデルの 1 段目のネットワークとして選択された。

表 4.2: データセット 1 に基づいてトレーニングした結果 (6 : 2 : 2 の割合でランダムに分割).

Network	mAP@0.5(%)	Recall(%)	Omission(%)
YOLOv5l	90.4	85.2	14.8
YOLOv5x	88.4	81.6	18.4
YOLOv7	88.3	83.8	16.2
YOLOv7-X	82.9	80.1	19.9
Faster R-CNN(R50-FPN)	80.9	74.0	26.0
Faster R-CNN(R101-FPN)	81.2	75.9	24.1
RetinaNet(R50)	78.4	72.6	27.4
RetinaNet(R101)	78.6	73.7	26.3

表 4.3: データセット 2 に基づいてトレーニングした結果 (日付による分割).

Network	mAP@0.5(%)	Recall(%)	Omission(%)
YOLOv5l	79.7	69.8	30.2
YOLOv5x	79.8	69.7	30.3
YOLOv7	77.7	72.4	27.6
YOLOv7-X	79.8	71.4	28.6
Faster R-CNN(R50-FPN)	74.0	73.1	26.9
Faster R-CNN(R101-FPN)	75.5	73.0	27.0
RetinaNet(R50)	73.6	72.9	27.1
RetinaNet(R101)	73.6	71.1	28.9

表 4.4: データセット 3 に基づいてトレーニングした結果 (場所による分割).

Network	mAP@0.5(%)	Recall(%)	Omission(%)
YOLOv5l	81.0	74.4	25.6
YOLOv5x	81.3	72.1	27.9
YOLOv7	82.4	73.7	26.3
YOLOv7-X	82.2	74.1	25.9
Faster R-CNN(R50-FPN)	81.3	72.0	26.9
Faster R-CNN(R101-FPN)	77.8	69.6	30.4
RetinaNet(R50)	77.4	71.3	28.7
RetinaNet(R101)	75.0	69.5	30.5



### 4.3.3 検証ネットワークの訓練と結果

最適なデータセットと物体検出ネットワーク YOLOv5l については 4.3.2 項で説明した通りである。次は適切な検証ネットワークを選ぶ。検証ネットワークは、分類を行うネットワークであるため、トレーニングするにはデータセットが必要となる。データセットを生成するために、YOLOv5l の信頼度の閾値を 0.1 に設定した。信頼度の閾値を 0.1 という低い値にすることにより、誤検出が多くなるため、正しく検出された画像に加え、誤検出された画像も多数得ることができる。

そして、データセット 1 のトレーニングセットとバリデーションセットをトレーニング済みの YOLOv5l に入力し、4473 枚のトマトの画像を切り出した。これらの切り出した画像には、尻腐れ果にかかっているトマトのような正しい検出と背景や健康なトマトなどの誤った検出が含まれていた。分類の学習を行うデータセットとして使用するために、切り出した画像は手動でそれぞれのフォルダに分類した。

尻腐れ果のトマトに比べて誤検出の数（「その他」）が少ないため、オーグメンテーション手法を使ってデータを拡張して数を増やした（表 4.5）。

表 4.5: 用いたオーグメンテーション手法の詳細。

方式	確率
左右反転	0.5
上下反転	0.5
ランダム明るさ	0.5
ランダムコントラスト	0.5
ランダム彩度	0.3
ランダム拡大	0.5
90 度回転	0.5

それから、すべての切り出しを 224×224 にリサイズし、トレーニングセット、バリデーションセット、テストセットに 6:2:2 の割合でランダムに分割した。表 4.6 に示すように、トレーニングセットに 5065 枚の画像が含まれ、バリデーションセットに 1595 枚の画像が含まれ、テストセットに 1597 枚の画像が含まれていた。尻腐れ果と「その他」の具体的な例は 4.9 に示す。

表 4.6: 分類ネットワークをトレーニングするためのデータセットの詳細。

	Training Set	Validation Set	Test Set
尻腐れ果	2701	787	818
その他	2364	808	779
Total	5065	1595	1597

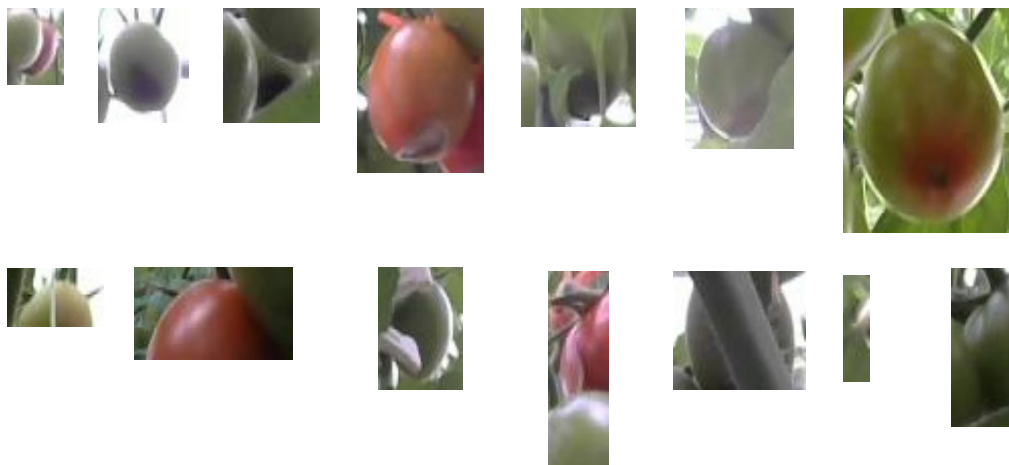


図 4.9: 分類ネットワークをトレーニングするためのデータの例。(上) 尻腐れ果；(下) その他.

優れたパフォーマンスの分類ネットワークを選択するために、ResNet [63], MobileNet, および、DenseNet [64] の一部のバージョンをトレーニングして比較した。ネットワークの出力は「尻腐れ果」と「その他」という2つのクラスに設定した。トレーニングのパラメータは公式の推奨するパラメーターを使用した。作成されたデータセットでネットワークをトレーニングした結果は表 4.7 に示す。MobileNetv2 は全体的なパフォーマンスが高かった。したがって、本研究では、MobileNetv2 を 2 段病気検出モデルの 2 段目のネットワークとして選択した。

表 4.7: 各分類ネットワークのトレーニング結果.

Network	Accuracy(%)	Size(MB)
ResNet18	96.7	42.7
ResNet50	95.9	90.0
MobileNetv2	96.7	8.8
MobileNetv3(large)	96.4	16.2
DenseNet121	97.6	27.1
DenseNet201	97.7	70.3

#### 4.3.4 2 段病気検出モデルの結果

本項で使用するテストデータはデータセット 1 のテストセットである。テストデータは 2 段病気検出モデルに入力され、パフォーマンスを確認した。

2 段病気検出モデルの最終的な出力はラベル付きの病気のトマトの切り抜いた画像である。

そのため、切り抜いた画像のラベルと座標に基づき、バウンディングボックスの形で元の画像データに還元することができる。テストデータのグラウンドトゥールズを利用し、IoU (Intersection over Union) に基づき正しい検出の数を推定した。式は次のとおりとなる。

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (4.2)$$

ここで、A はグラウンドトゥールズのバウンディングボックスの面積を表し、B はモデルにより予測されたバウンディングボックスの面積を表す。IoU の値は 0 から 1 の間となり、1 に近いほど予測のバウンディングボックスとグラウンドトゥールズのバウンディングボックスの重なりが大きくなる。例えば、Pascal VOC 2008 [65] では、IoU の閾値が 0.5 を超えると正しい予測と見なされている。本研究では、将来提案のシステムを実用化する可能性があると考えているため、実験では IoU の閾値は 0.65 に厳しく設定した。

正しい予測の数に基づいて、偽陽性と偽陰性の数を取得できる。よって、偽陽性と偽陰性の割合を計算できる。計算は次のように記述できる。

$$False\ Positive\ Rate = \frac{N_p - N_{cp>0.65}}{N_p} \quad (4.3)$$

$$False\ Negative\ Rate = \frac{N_{gt} - N_{cp>0.65}}{N_{gt}} \quad (4.4)$$

ここで、 $N_p$  は予測の総数を表し、 $N_{cp>0.65}$  は IoU の閾値が 0.65 より大きい場合の正しい予測の総数を表す。 $N_{gt}$  はグラウンドトゥールズの総数を表す。

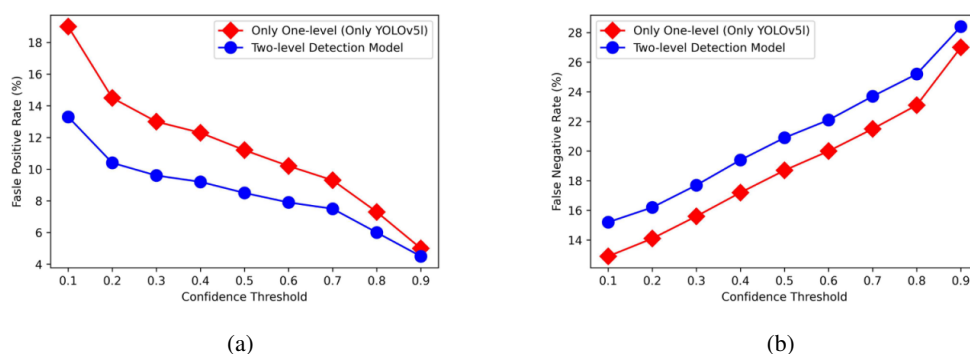


図 4.10: 異なる信頼度の閾値で 2 段階検出モデルと 1 段階目のネットワーク (YOLOv5l) のみを実行した結果。(a) 偽陽性率の比較; (b) 偽陰性率の比較。

2 段階検出モデルの YOLOv5l の出力は信頼度の閾値が異なると変わるため、閾値は 0.1 から 0.9 まで 0.1 ごとに実行した。検出の結果によって偽陽性率と偽陰性率を計算した。本研究では尻腐れ果のみに注目し、モデルは尻腐れ果にかかっているトマトを含む画像を使用してトレーニングされたため、各信頼度の閾値で偽陽性率と偽陰性率が 1 つの値のみが取得し

た。比較のため、2段病気検出モデルと1段目のみ（YOLOv5lのみ）の場合の実行結果をそれぞれ計算した。最終結果は図 4.10 に示す。

図 4.10 に示すように、1段目のネットワークのみまたは2段病気検出モデルにかかわらず、検出ネットワークの信頼度の閾値が増加するにつれて偽陽性率は減少した。偽陰性率は信頼度の閾値と共に増加した。信頼度の閾値が 0.1 の時、2段病気検出モデルの偽陽性率は 13.3%、偽陰性率は 15.2%であった。

病気のトマトをできるだけ多く検出し、見落としをできるだけ減らすため、当システムでは信頼度の閾値を 0.1 に設定する。図 4.11 に示すように、信頼度の閾値を 0.1 に設定すると、YOLOv5lのみを使用した場合にいくつかの誤った結果が検出された。図 4.11(a) と (c) の矢印が指しているように、健康なトマトの一部が尻腐れ果にかかっているトマトとして誤って検出された。図 4.11(e) では、バックグラウンド（白いクランプ）が尻腐れ果にかかっているトマトとして誤って検出された。図 4.11(b), (d), (f) に示すように、2段階検出モデルを使用すると、誤検出が除去された。ただし、すべての誤検出が正しく分類されたわけではなく、図 4.11(a) の赤い矢印で示された誤検出は、図 4.11(b) にもあった。

さらに、信頼度の閾値を 0.1 に設定した場合、2段病気検出モデルの分類ネットワークの混同行列を表 4.8 に示す。この値に基づいて、MobileNetv2 のトレーニング結果に近い精度（96.4%）を得られることがわかる。

表 4.8: 信頼度の閾値が 0.1 に設定された場合、2段病気検出モデルの分類ネットワークの混同行列。

		Actual Values	
		尻腐れ果	その他
Predicted Values	尻腐れ果	909	16
	その他	21	80

## 4.4 システムの結果

提案の病気モニタリングシステムの有効性を実証するために、トレーニングに使用されていたデータを収集した以外の日によって収集した3日間の画像データを使用した。近い日付のデータは結果が類似する可能性があるため、検証に用いたデータは間隔の広い3日（4月10日、5月12日、6月10日）を選択した。

垂直方向の距離はマーカなどを使って測ることはできないので、Z軸の座標は次のように定義した。最も下のネスタブルモジュール（L1）のカメラが撮影した画像が0~100、L2のカメラが撮影した画像が100~200、L3のカメラが撮った画像は200~300の値を与えて、各画像の間隔は10に定義した。たとえば、L1によって収集された2番目の画像に尻腐れ果にかかっているトマトが含まれている場合、Z軸座標は10にする。同じ画像に複数の検出結果があった場合、重ねるようにした。結果は図 4.12 に示す。赤または緑の点は、システムが予測

Detection Results of  
One-level (Only YOLOv5l)



Detection Results of  
Two-level Model



図 4.11: 信頼度の閾値が 0.1 に設定された場合の検出結果の例。2 段病気検出モデルは青い矢印が指している誤検出を削除したが、赤い矢印が示した誤検出は削除できなかった。

した尻腐れ果が発生している場所を表す。濃い色はその位置に複数の尻腐れ果のトマトがあることを示す。

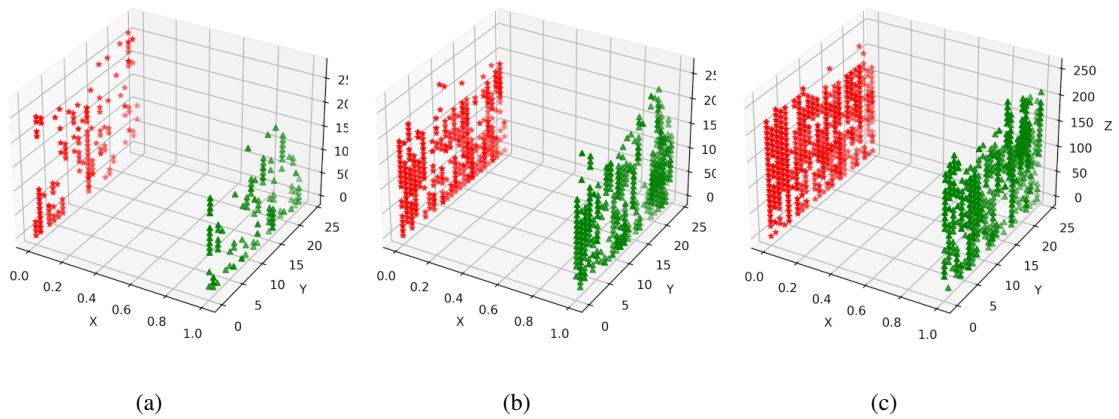


図 4.12: システムによって報告された尻腐れたトマトの分布。赤い点は列  $x_0$  の結果で、緑色の点は列  $x_1$  の結果である。より濃い点は複数の病気のトマトが重なっていることを意味する。(a)4月10日の結果；(b)5月12日の結果；(c)6月10日の結果。

## 4.5 ディスカッション

### 4.5.1 システムの実装結果について

実験を行った温室では、3月の末にトマトの栽培が始まった。最初は尻腐れ果の発生は少ないが、温室の温度や湿度などの制御をしない限り<sup>8</sup>、尻腐れ果が発生していた。図 4.12 から、4月10日は尻腐れ果にかかっているトマトが一番少なく、5月12日は多くなり、6月10日はより多いことが分かった。これは実際に目視でおよそ確認した状況と一致している。提案システムにより尻腐れ果実の検出が完全に行えるわけではないが、全体的な発生状況を把握することは可能であり、提案システムの検出率が達成されていれば栽培条件の制御につなげることができるため、病気への対処が早期に行えるという意味では有効であることを確認した。

### 4.5.2 データセットの分割とネットワークの比較

本研究での実験では、データセットの分割方法を3種類のデータセットを作成し、結果を比較した。すなわち、一定な割合でのランダムな分割、日付による分割、および、場所による分割の3つである。これらのデータセットでさまざまな物体検出ネットワークをトレーニ

<sup>8</sup>実験を行ったトマトは、開発途中の実験品種のトマトであるため、有効な栽培条件は見つかっていない。

ングした結果、YOLOv5lを一定な割合でランダムに分割したデータセットを用いて学習した場合に最良の結果を示した。

原因としては、ランダムに分割するというのは、天気や時間、環境光をランダムに各セット（トレーニングセット、バリエーションセット、テストセット）に配分することに相当するため、均一ではないが、YOLOv5lがさまざまな状態を学習でき、汎用的な性能が獲得できたと考えられる。

日付で分割したデータセットを用いたトレーニングでは、mAP@0.5の値が80%を超えるネットワークはなかった。この原因の1つに、異なる日付で収集された画像データに含まれている病気のトマトは、病変の程度が異なることが挙げられる。また、天気の違いでトマトの表面特徴も変わることが考えられる。さらに、撮影時期の違いにより、トマトの大きさと色が異なることも原因の1つとして挙げられる。結果的に、ネットワークの汎化パフォーマンスが低下していることがわかった。

直観的には、場所による分割で作成したデータセットを用いたトレーニングにより、ネットワークがすべての場所から尻腐れたトマトの特徴を学習できることが予想される。しかしながら、場所による分割では、ランダム分割よりもはるかにパフォーマンスが低下した。考えられる原因の1つは、ユーザが時々トマトを調整したりまたは再栽培したりしていたため、日付間隔が長すぎると、ネットワークがすべての場所から病気のトマトの特徴を学習できないことが挙げられる。

さらに、分類ネットワークは深い大規模なネットワークほど良い性能を持つとは限らない。実験で最良の結果を達成したMobileNetv2はResNet50よりも精度が高いが、サイズは比較的小さい。

#### 4.5.3 2段階病気検出モデルについて

図4.13の赤い矢印が示すように、2段階病気検出モデルには失敗の検出があった。図4.13(a)の場合、健康のトマトを尻腐れたトマトとして検出した。考えられる原因として、その健康のトマトの先端が他のトマトのへたと被っており、先端部分が黒く見えるためである。このように、本物の尻腐れ果と見た目が類似しているため、結果的に提案手法で誤検出した。

図4.13(b)の検出結果は尻腐れ果であるかどうか、解像度の原因で人でも判断がつかないものである。先端部分が黒いという特徴から検出されたが、正しい検出として認めるのは難しいと考える。

図4.13(c)は、葉の間の影を尻腐れ果として検出した例である。図4.13(d)も同じく黒っぽい健康のトマトを尻腐れたトマトとして検出した。異常をまとめると、提案の2段階病気検出モデルは尻腐れ果の特徴を識別できるが、似ているものは誤検出する。似ているものは先端部分が黒くて、本当の尻腐れ果と特徴が近い。

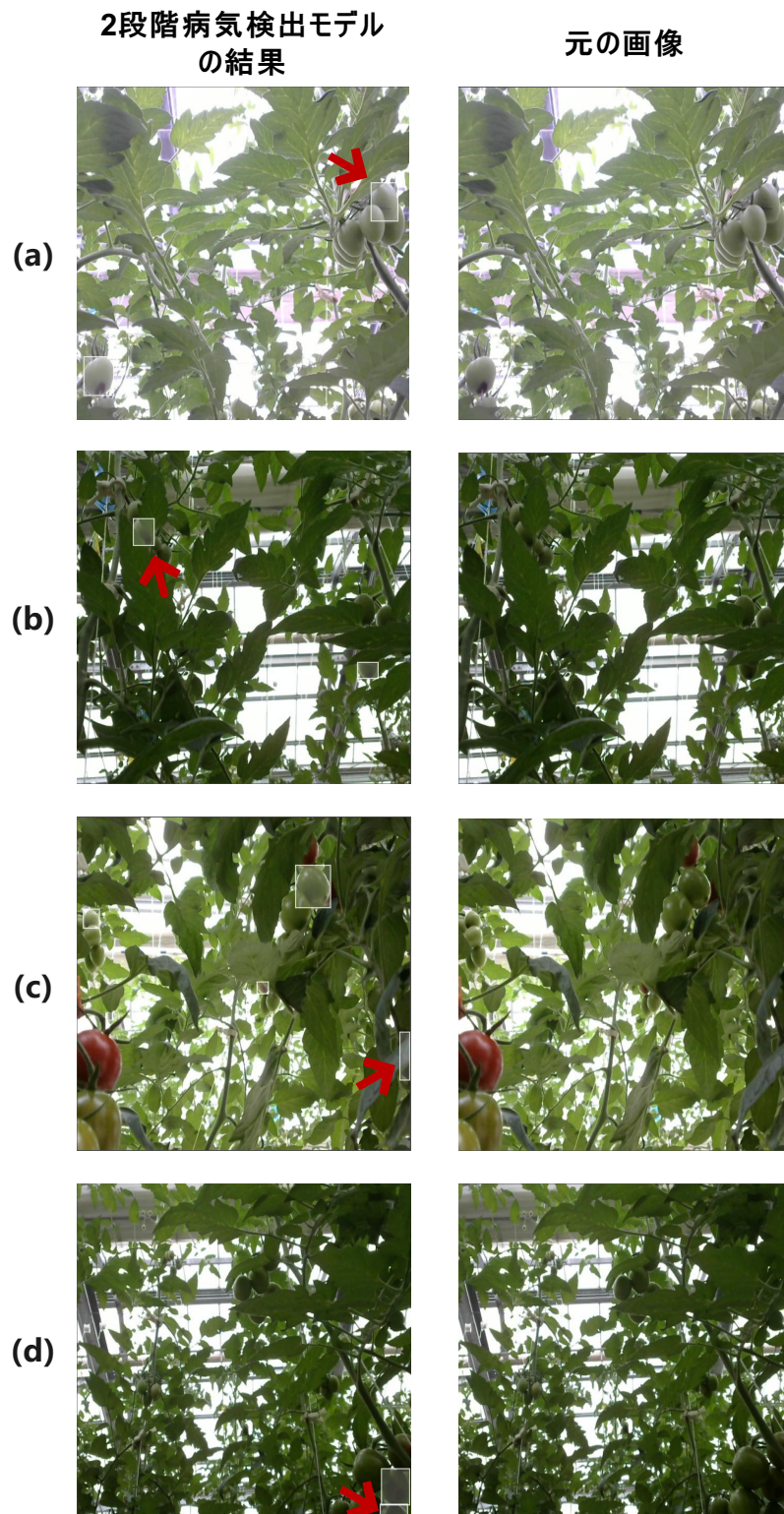


図 4.13: 提案の 2 段階病気検出モデルの失敗例 (赤い矢印が指しているところ). (a) 失敗の検出例; (b) 元の画像.



#### 4.5.4 本研究の限界と今後の課題

まず、本論文を執筆する時点で、提案のシステムのモニタリングは温水パイプでの自動実行を優先して実装した。プロトタイプロボットを迅速に開発するために、ロボットはエンコーダとLiDARなどのセンサを省略している。しかし、これはロボットの全自動化を制限する結果となり、ロボットを操縦して温水パイプの頭までに移動させることも不便であった。また、目視検査エリアを設置しているが、人の介入を必要としないシステムが望まれている。

次に、提案のシステムは収集された画像から尻腐れたトマトのみを検出し、他の病気に対する実験は行われていない。よって、複数のトマトの病気に対するモニタリングの有効性は確認できない。

また、提案した2段階病気検出モデルは実験を行った温室のトマトのデータのみを用いて学習しているため、別の温室のトマトに対応するためには転移学習が必要であると考えられる。トマトの種類の違いや温室の環境の違い、地域によって太陽光の入射角が変わるとトマトの表面特徴も変わることなどが原因で、現在学習に使っているデータのみではその他の温室のトマトに対応することは困難である。適用する現場で、ある程度の量の画像データを収集する必要がある。

## 第5章 結論

### 5.1 まとめ

本研究では、温室においてトマトの病気モニタリングシステムを提案した。このシステムは、まず本研究で開発した新型モジュール式且伸縮可能なロボットを使用して温室で作業して画像を収集する。収集した画像データはサーバに送信される。それからサーバでは2段の病気検出モデルが実行されて病気を検出する。最終的に、ユーザは病気のトマトの場所を把握できるため、タイムリに対処することより損失を減らすことができる。

### 5.2 研究の成果と貢献

本研究による成果および貢献は主に3つがある。

#### 5.2.1 新型ロボットの提案

温室で栽培されているトマトは高い位置にガイドされることが多いが、どの程度の高さまでガイドされているかは環境による。監視のためにトマトの大部分を過不足なくモニタリングできるロボットは存在していなかった。本研究で提案した新型ロボットは伸縮可能なデザインを持っており、温室においてモニタリングすべきトマトのある高さに合わせてデータ収集するための高さの設定が可能である。そして、新型ロボットはモジュール式に設計されているため、ユーザのニーズ通りに配置することができる。すなわち、コストをコントロールすることができる。また、プロトタイプの実機の長い間に実際に温室で稼働していた実績から見ると、新型ロボットは高い安定性を持っている。以上のメリットから、提案の新型ロボットは温室においてモニタリング用ロボットとして農業に広げられると考えられる。

#### 5.2.2 誤検出率の減少方法の提案

どの物体検出ネットワークでも予測ミスや失敗がある。誤った予測については、本研究では物体検出ネットワークの出力をさらに検証する設計により、誤った予測を減らす2段病気検出モデルを提案したことで誤った予測を減らした。提案手法により、ユーザは多く作業時間を節約ことができ、生産性を向上できると考える。

### 5.2.3 トマト病気モニタリングの自動化の実現

トマトの市場は非常に大きいですが、温室においてトマトの病気の自動モニタリングに関する研究は少ない。提案のシステムは自律行動できる新型ロボットと自動的に病気を検出するプログラムを動かすサーバを用いることで、病気のトマトの場所をタイムリにユーザに通知しことを実現した。このシステムは温室内のトマトの病気を最小限の人工支援で自動的にモニタリングし、人工的や時間的なコストを減らすことができると考えられる。

## 5.3 他の作物への可能性の検討

トマトの他に、イチゴやキュウリなどさまざまな作物が温室栽培になっている。本研究が提案したシステムはトマトをモニタリングすることができるが、他の作物もモニタリングできると考える。提案の新型ロボットはさまざまな高さに対応できるだけでなく、コストもコントロール可能で、栽培されている作物の特徴に対して柔軟に配置できる。特に粒型の作物はトマトの果実と似ているため、提案の2段階病気検出モデルも適用すると考える。

## 5.4 今後の課題と展望

本研究では、温室において新型モジュール式且伸縮可能なロボットを使用するトマトの病気モニタリングシステムを実現したが、まだ実用化に向けた第1段階であり、将来の課題は多く残されている。

まず、プロトタイプのロボットを迅速に開発するために、エンコーダやライダーセンサなどは省略された。しかし、これはロボットの機能を制限し、結果的にロボットを全自動に開発することが不可能となる要因である。そして、ユーザがロボットを遠隔操作して温水パイプまで移動させることも不便で、効率的ではないことは明らかである。したがって、ロボットの稼働効率を高めるために、必要なセンサなどを追加し、完全に自動的に働くアルゴリズムを開発することが今後の重要な課題となる。

また、本研究で提案したロボットは同時に6台のカメラを上げたり下げたりしてカメラの位置を調整することができるが、カメラの向きを自動的に調整することができない。事前にカメラの角度を設置する必要がある。要するに、トマトの先端をとればへたをとることができなくなる。将来の課題として、トマトに関する全般をとるためにモータなどを追加し、カメラの向きを調整できるようにする必要性が生じる可能性がある。

このシステムでは、収集された画像から尻腐れ果のトマトのみが検出され、他の病気に対する実験は行われていない。そのため、複数の病気のモニタリングに対して、提案のシステムの有効性は確認していない。今後の課題としては、複数の病気のモニタリングを行い、提案のシステムの効果を確かめることが挙げられる。また、実用化のためには人間による操作をできるだけ減らし、さらに人間の作業を邪魔することのないシステムが望まれている。検出率と誤検出率をできるだけ減らすことも今後の不可欠な課題となる。

また、当システムが収集した画像データには、トマトの位置と時間の情報が含まれるため、トマトの時間的な変化を含むデータから成長状況などを推測できると考える。よって、もう1つの課題として、収集したデータに基づき、トマトの生育状態などをデジタル化し、より正しい生育状態や生産量などの予測を行うことも挙げられる。

## 参考文献

- [1] 農文協. トマト大事典. 農山漁村文化協会, 2015.
- [2] statista. Global Production of Vegetables in 2021, by Type (in Million Metric Tons). <https://www.statista.com/statistics/264065/global-production-of-vegetables-by-type/>, (accessed on 20 January 2023).
- [3] エペ・フューヴェリンク (編著); 中野明正, 東出忠桐, 松田怜 (監訳). トマト 100 トンどりの新技術と理論 – 低投入多収をめざして. 農山漁村文化協会, 2020.
- [4] 田淵俊人. まるごとわかるトマト. 誠文堂新光社, 2017.
- [5] M. Viuda-Martos, E. Sanchez-Zapata, E. Sayas-Barberá, E. Sendra, J. A. Pérez-Álvarez, and J. Fernández-López. Tomato and Tomato Byproducts. Human Health Benefits of Lycopene and Its Application to Meat Products: A Review. *Critical Reviews in Food Science and Nutrition*, Vol. 54, No. 8, pp. 1032–1049, 2014.
- [6] 農業・食品産業技術総合研究機構. 心止まり – ルーラル電子図書館 – 農業技術事典 NAROPEDIA. <http://lib.ruralnet.or.jp/nrpd/#koumoku=12653>, (accessed on 20 January 2023).
- [7] 周長吉. 現代温室工程 (中国語). 化学工業出版社, 2010.
- [8] Gleason, M.L. and Edmunds, B.A. (2006) *Tomato Diseases and Disorders*. University Extension PM 1266, Iowa State University, Ames.
- [9] A. Esteva, B. Kuprel, R. Novoa, J. Ko, S. Swetter, H. Blau, and S. Thrun. Dermatologist-level Classification of Skin Cancer with Deep Neural Networks. *Nature*, Vol. 542, pp. 115118, 2017.
- [10] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J Big Data*, Vol. 8, No. 1, p. 53, 2021.
- [11] Xuewei Wang and Jun Liu. Tomato Anomalies Detection in Greenhouse Scenarios Based on YOLO-Dense. *Front. Plant Sci.*, Vol. 12, p. 634103, April 2021.

- [12] Mubashiru Olarewaju Lawal. Tomato Detection Based on Modified YOLOv3 Framework. *Sci Rep*, Vol. 11, No. 1, p. 1447, December 2021.
- [13] Jun Liu and Xuewei Wang. Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. *Front. Plant Sci.*, Vol. 11, p. 898, June 2020.
- [14] Dr. M. Sunil Kumar Dr. V. Anantha Natarajan, Ms. Macha Babitha. Detection of Disease in Tomato Plant Using Deep Learning Techniques. *International Journal of Modern Agriculture*, Vol. 9, pp. 525–540, Dec. 2020.
- [15] Yang Zhang, Chenglong Song, and Dongwen Zhang. Deep Learning-Based Object Detection Improvement for Tomato Disease. *IEEE Access*, Vol. 8, pp. 56607–56614, 2020.
- [16] Siti Zulaikha Muhammad Zaki, Mohd Asyraf Zulkifley, Marzuraikah Mohd Stofa, Nor Azwan Mohammed Kamari, and Nur Ayuni Mohamed. Classification of Tomato Leaf Diseases Using MobileNet v2. *IJ-AI*, Vol. 9, No. 2, p. 290, June 2020.
- [17] Ding Jiang, Fudong Li, Yuequan Yang, and Song Yu. A Tomato Leaf Diseases Classification Method Based on Deep Learning. In *2020 Chinese Control And Decision Conference (CCDC)*, pp. 1446–1450, Hefei, China, August 2020. IEEE.
- [18] Tao Lu, Baokun Han, Lipin Chen, Fanqianhui Yu, and Changhu Xue. A Generic Intelligent Tomato Classification System for Practical Applications Using DenseNet-201 with Transfer Learning. *Sci Rep*, Vol. 11, No. 1, p. 15824, December 2021.
- [19] Prajwala Tm, Alla Pranathi, Kandiraju SaiAshritha, Nagaratna B. Chittaragi, and Shashidhar G. Koolagudi. Tomato Leaf Disease Detection Using Convolutional Neural Networks. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–5, Noida, August 2018. IEEE.
- [20] Shengyi Zhao, Yun Peng, Jizhan Liu, and Shuo Wu. Tomato Leaf Disease Diagnosis Based on Improved Convolution Neural Network by Attention Module. *Agriculture*, Vol. 11, No. 7, p. 651, July 2021.
- [21] Amreen Abbas, Sweta Jain, Mahesh Gour, and Swetha Vankudothu. Tomato Plant Disease Detection Using Transfer Learning with C-GAN Synthetic Images. *Computers and Electronics in Agriculture*, Vol. 187, p. 106279, August 2021.
- [22] Mark Ramaker, Antoon H. Boode, Cock Heemskerk, and Lucien Fesselet. Accurate UAS Flight inside a Greenhouse a Novel Algorithm Combining Sparse Block Matching Optical Flow with UWB Localization. In *2020 21st International Conference on Research and Education in Mechatronics (REM)*, pp. 1–6, Cracow, Poland, December 2020. IEEE.

- [23] Linlu Zu, Yanping Zhao, Jiuqin Liu, Fei Su, Yan Zhang, and Pingzeng Liu. Detection and Segmentation of Mature Green Tomatoes Based on Mask R-CNN with Automatic Image Acquisition Approach. *Sensors*, Vol. 21, No. 23, p. 7842, November 2021.
- [24] Dasom Seo, Byeong-Hyo Cho, and Kyoung-Chul Kim. Development of Monitoring Robot System for Tomato Fruits in Hydroponic Greenhouses. *Agronomy*, Vol. 11, No. 11, p. 2211, October 2021.
- [25] Yuhao Ge, Sen Lin, Yunhe Zhang, Zuolin Li, Hongtai Cheng, Jing Dong, Shanshan Shao, Jin Zhang, Xiangyu Qi, and Zedong Wu. Tracking and Counting of Tomato at Different Growth Period Using an Improving YOLO-Deepsort Network for Inspection Robot. *Machines*, Vol. 10, No. 6, p. 489, June 2022.
- [26] Patrick Wspanialy and Medhat Moussa. Early Powdery Mildew Detection System for Application in Greenhouse Automation. *Computers and Electronics in Agriculture*, Vol. 127, pp. 487–494, September 2016.
- [27] Xiaoliang Zhou, Pengbo Wang, Guanglin Dai, Jiawen Yan, and Zhan Yang. Tomato Fruit Maturity Detection Method Based on YOLOV4 and Statistical Color Model. In *2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 904–908, Jiaxing, China, July 2021. IEEE.
- [28] Hubert Fonteijn, Many Afonso, Dick Lensink, Marcel Mooij, Nanne Faber, Arjan Vroegop, Gerrit Polder, and Ron Wehrens. Automatic Phenotyping of Tomatoes in Production Greenhouses Using Robotics and Computer Vision: From Theory to Practice. *Agronomy*, Vol. 11, No. 8, p. 1599, August 2021.
- [29] Gan DU. Research Status of Lifting Mechanism of Lifting Platform (in Chinese). *MECHANICAL ENGINEERING & AUTOMATION*, No. 2, pp. 205–207, April 2013.
- [30] Chengcheng Xu, Jiping Zhou, Huacheng An, Mianshun Zhu, and Zhonglin Xu. Design and Simulation Analysis of the Lifting Mechanism of the Experimental Waste Liquid Barrel Handling Robot (in Chinese). *Machine Design and Manufacturing Engineering*, Vol. 51, No. 11, pp. 45–48, November 2022.
- [31] Yu-jie SUN and Shan-wen ZHANG. Design of Leaf Picking Device (in Chinese). *MECHANICAL ENGINEERING & AUTOMATION*, No. 2, pp. 99–100, April 2021.
- [32] 曾午平, 衛良保. 油圧シザーリフトプラットフォームの設計と計算 (訳名, 中国語). *Hoisting and Conveying Machinery*, No. 1, pp. 20–22, 2010.
- [33] 張宣政. *The Design and Research about the Automatic Inventory Smart Car in Stack Room (in Chinese)*. 華東理工大学, 2016.

- [34] Jiao Wang, Yinwei Wang, and Xingshuo Wang. 立体倉庫スタッカの設計と研究 (訳名, 中国語) . *Machinery and Equipment Research & Development*, No. 11, pp. 131–133, June 2022.
- [35] Tao XIN, Youping WANG, and Shiqing ZHOU. Development of New-Model In-Warehouse Mast Type Self-Propelled Lifter (in Chinese). *Construction Machinery Technology & Management*, No. 9, pp. 86–88, September 2007.
- [36] 辛濤, 王幼平, 艾会莉. マスト式昇降機構の設計と計算 (訳名, 中国語) . 起重運輸機械, No. 4, pp. 18–23, April 2010.
- [37] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017.
- [38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.
- [39] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.
- [40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016. IEEE.
- [41] Ultralytics. Yolov8. <https://github.com/ultralytics/ultralytics>, (accessed on 20 January 2023).
- [42] Ultralytics. Yolov5. <https://github.com/ultralytics/yolov5>, (accessed on 20 January 2023).
- [43] WZMIAOMIAO. Yolov5l. <https://github.com/ultralytics/yolov5/issues/6998>, (accessed on 20 January 2023).
- [44] ROS. <https://www.ros.org/>, (accessed on 20 January 2023).
- [45] 岡田浩之. ロボットプログラミング ROS2 入門. 科学情報出版株式会社, 2020.
- [46] Piotr Skalski. Make Sense. <https://www.makesense.ai> (accessed on 20 January 2023).
- [47] AUTODESK. Fusion 360. <https://www.autodesk.co.jp/products/fusion-360/overview?term=1-YEARtab=subscription> (accessed on 20 January 2023).
- [48] UltiMaker Cura. <https://ultimaker.com/ja/software/ultimaker-cura> (accessed on 20 January 2023).



- [49] EasyEDA. <https://easyeda.com/> (accessed on 20 January 2023).
- [50] MM Peet, GWH Welles, et al. Greenhouse Tomato Production. *Crop production science in horticulture*, Vol. 13, p. 257, 2005.
- [51] Yue Mu, Tai-Shen Chen, Seishi Ninomiya, and Wei Guo. Intact Detection of Highly Occluded Immature Tomatoes on Plants Using Deep Learning Techniques. *Sensors*, Vol. 20, No. 10, p. 2984, May 2020.
- [52] M. Benavides, M. Cantón-Garbín, J. A. Sánchez-Molina, and F. Rodríguez. Automatic Tomato and Peduncle Location System Based on Computer Vision for Use in Robotized Harvesting. *Applied Sciences*, Vol. 10, No. 17, p. 5887, August 2020.
- [53] Tongyu Xu, Xiangyu Qi, Sen Lin, Yunhe Zhang, Yuhao Ge, Zuolin Li, Jing Dong, and Xin Yang. A Neural Network Structure with Attention Mechanism and Additional Feature Fusion Layer for Tomato Flowering Phase Detection in Pollination Robots. *Machines*, Vol. 10, No. 11, p. 1076, November 2022.
- [54] Qingchun Feng, Wei Zou, Pengfei Fan, Chunfeng Zhang, and Xiu Wang. Design and Test of Robotic Harvesting System for Cherry Tomato. *International Journal of Agricultural and Biological Engineering*, Vol. 11, No. 1, pp. 96–100, 2018.
- [55] Pei Cao, Tianchi Wang, Liang Zhai, Shi' ao Niu, Li Liu, and Yinggang Shi. Design of 6-DOF Tomato Picking Lifting Platform. *Agriculture*, Vol. 12, No. 11, p. 1945, November 2022.
- [56] 大山光男. トマトの尻腐れ果発生要因に関する研究. 岡山大学大学院, 2017.
- [57] kaggle. Tomato Detection. <https://www.kaggle.com/datasets/andrewmvd/tomato-detection> (accessed on 20 January 2023).
- [58] Laboro. Laboro Tomato: Instance Segmentation Dataset. <https://github.com/laboroai/LaboroTomato> (accessed on 20 January 2023).
- [59] M Math, RajinderKumar; V. Dharwadkar, Dr. Nagaraj. Real-world Tomato Image Dataset for Deep Learning and Computer Vision Applications Involving Precision Agriculture, 2020. Mendeley Data, V1.
- [60] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors, 2022.
- [61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 39, No. 6, pp. 1137–1149, June 2017.

- [62] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Venice, October 2017. IEEE.
- [63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016. IEEE.
- [64] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.
- [65] Derek Hoiem, Santosh K Divvala, and James H Hays. Pascal VOC 2008 Challenge. *World Literature Today*, 2009.

## 謝辞

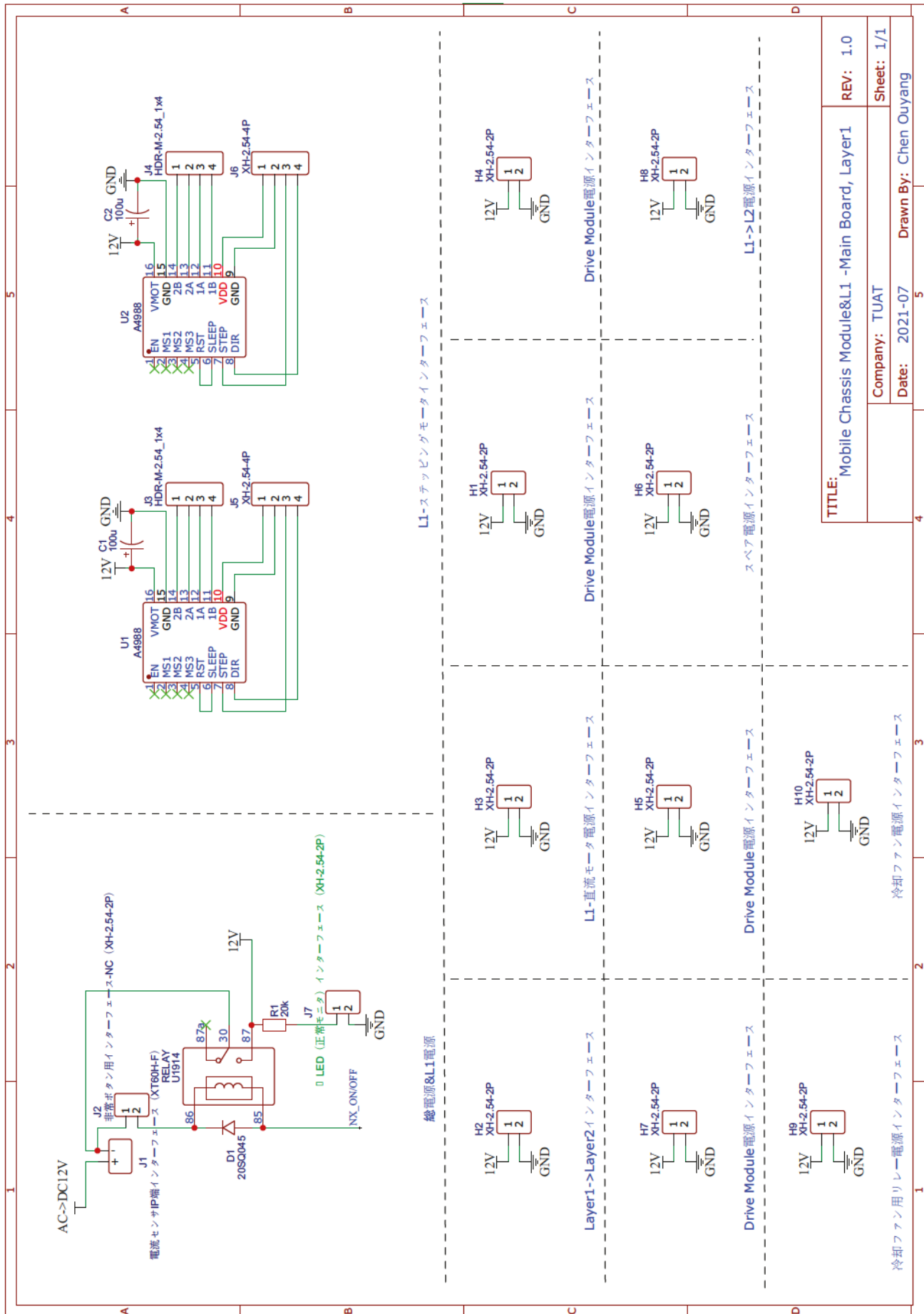
本研究の遂行にあたり，指導教員としてご指導いただいた清水郁子准教授に感謝いたします。学位論文審査では，齋藤教授，中條教授，近藤教授，堀田准教授に副査として貴重なご意見を賜って頂き，感謝いたします。そして，研究室のメンバであり，横河電機株式会社の社員である共同研究者の初谷恵美子様にご助言，ご協力いただきました。ここに感謝の意を表します。また，タキイ種苗株式会社の東将志様に温室においてロボットを操作したりして頂いていました。お礼を申し上げます。最後に，研究資金を提供いただいた横河電機株式会社に深く感謝申し上げます。

## 付録A 回路図

ロボットのコストと重さを減らすため、実装の際に、モバイルシャーシモジュールの上部と一番下のネスタブルモジュール (L1) の下部を統合した。よって、本来モバイルシャーシモジュールと L1 に別々に装着すべき回路はモバイルシャーシモジュールにまとめてインストールした。ユニバーサル基板のサイズは有限のため、回路を分割してメインボードとドライブモジュールボードに実装した。さらにメインボードにレイヤ 1 とレイヤ 2 が含まれている。実物は図 3.3 に示す。

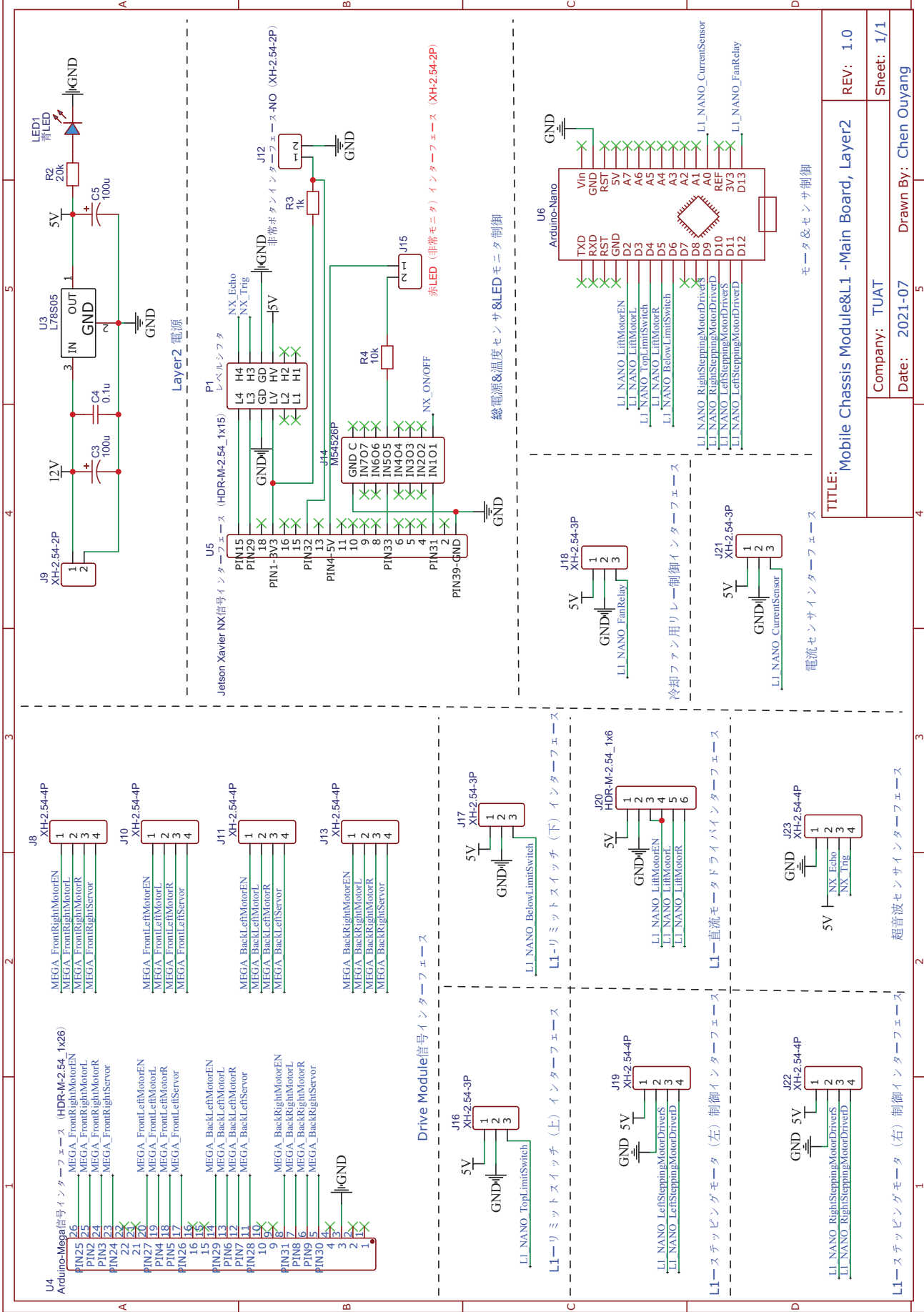
また、中間のネスタブルモジュール (L2) と一番上のネスタブルモジュール (L3) の回路は設計の通りにそれぞれ内蔵した。

## A.1 モバイルシャーシモジュールとL1の回路図-メインボード,レイヤ 1



TITLE: Mobile Chassis Module&L1 -Main Board, Layer1		REV: 1.0
Company: TUAT	Sheet: 1/1	
Date: 2021-07	Drawn By: Chen Ouyang	

## A.2 モバイルシャーシモジュールとL1の回路図-メインボード,レイヤ 2



TITLE: Mobile Chassis Module&L1 - Main Board, Layer2		REV: 1.0
Company: TUAT	Date: 2021-07	Sheet: 1/1
Drawn By: Chen Ouyang		

Layer2 電源

Drive Module信号インターフェース

総電源&温度センサ&LEDモニタ制御

モータ&センサ制御

超音波センサインターフェース

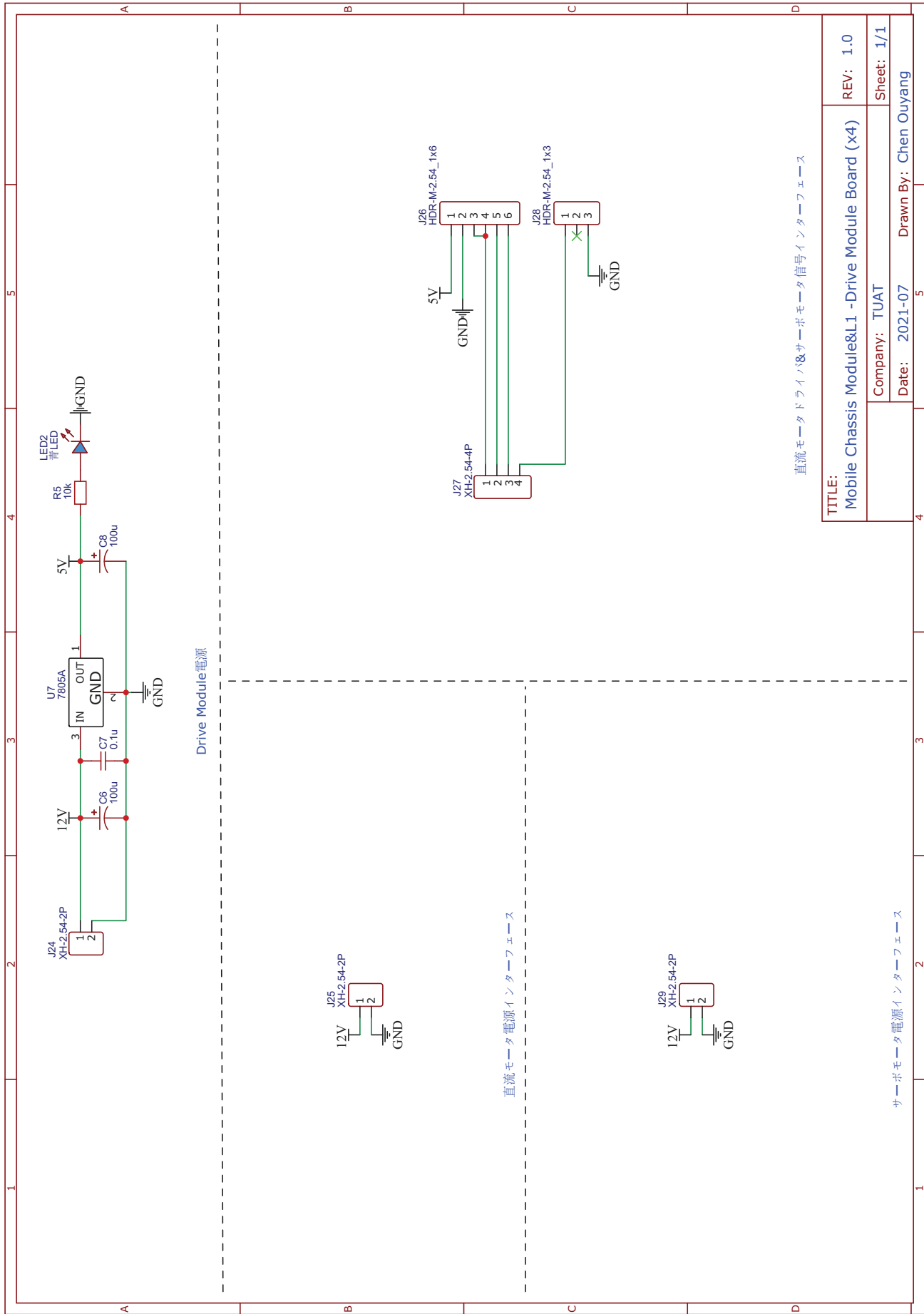
L1-ステッピングモータ (右) 制御インターフェース

冷却ファン用リレー制御インターフェース

電流センサインターフェース



### A.3 モバイルシャーシモジュールとL1の回路図-ドライブモジュール ボード



直流モータードライバ&サーボモーター信号インターフェース

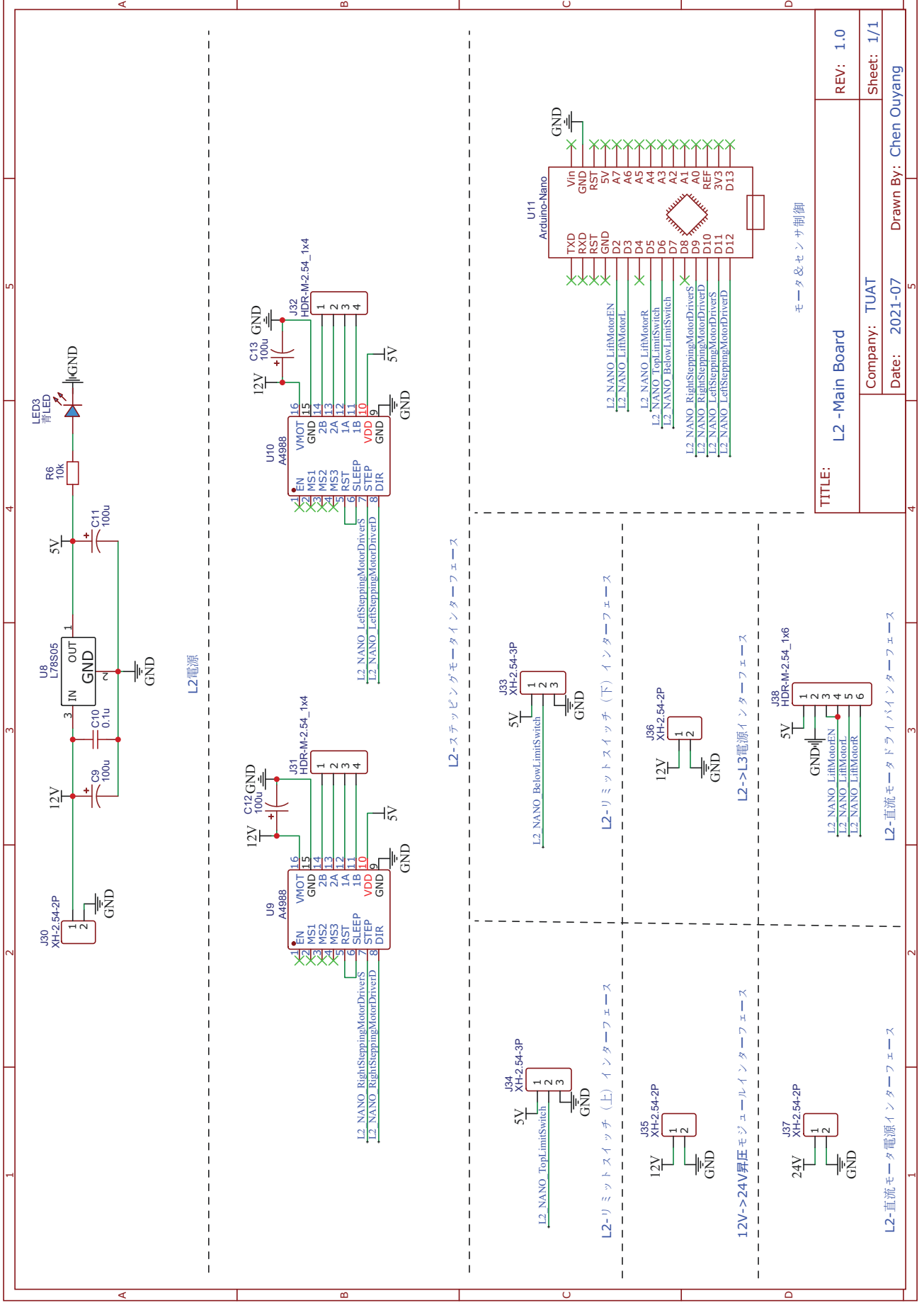
TITLE: Mobile Chassis Module&L1 - Drive Module Board (x4)		REV: 1.0
Company: TUAT	Sheet: 1/1	
Date: 2021-07	Drawn By: Chen Ouyang	

サーボモーター電源インターフェース

直流モーター電源インターフェース

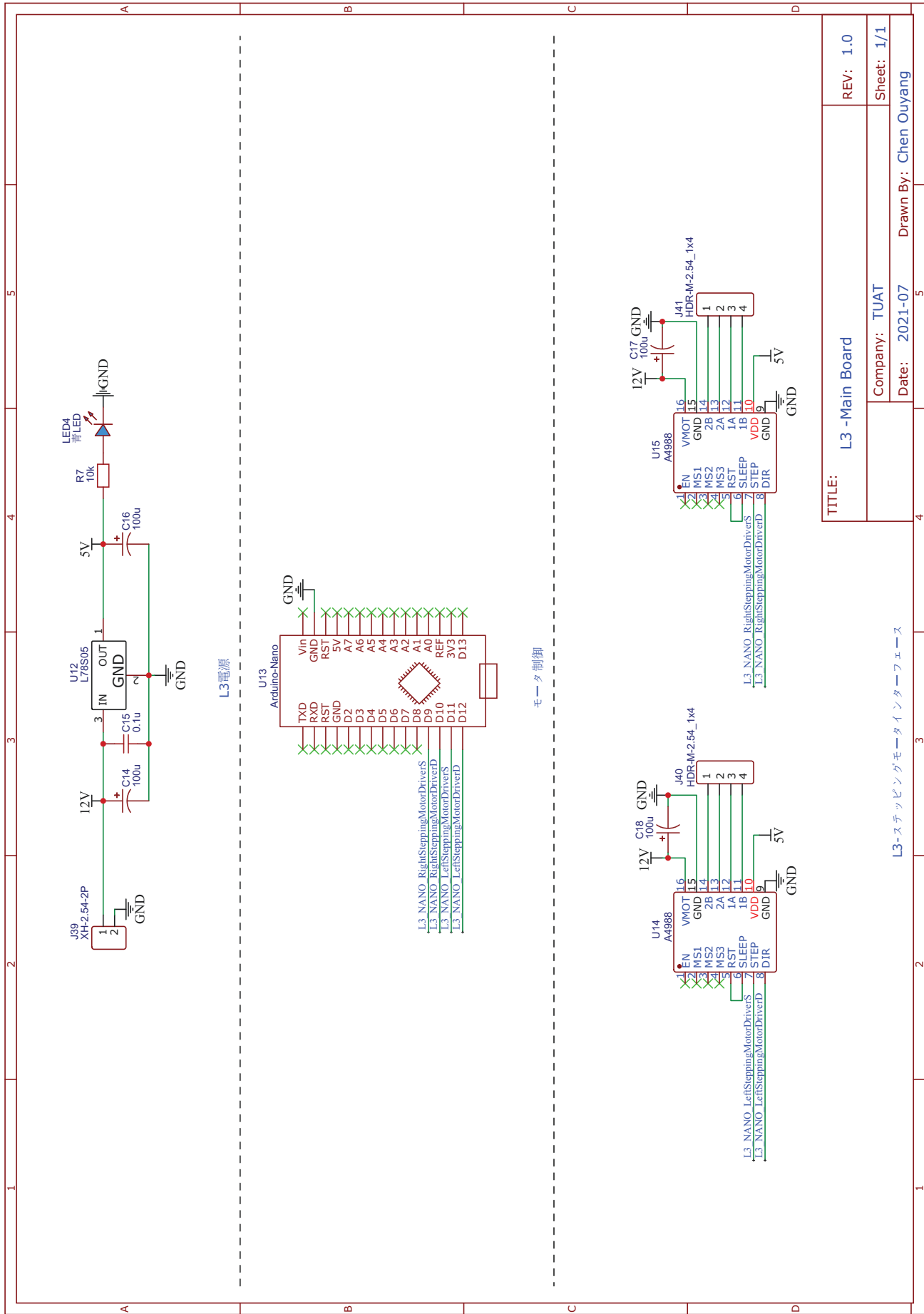
Drive Module電源

## A.4 L2の回路図

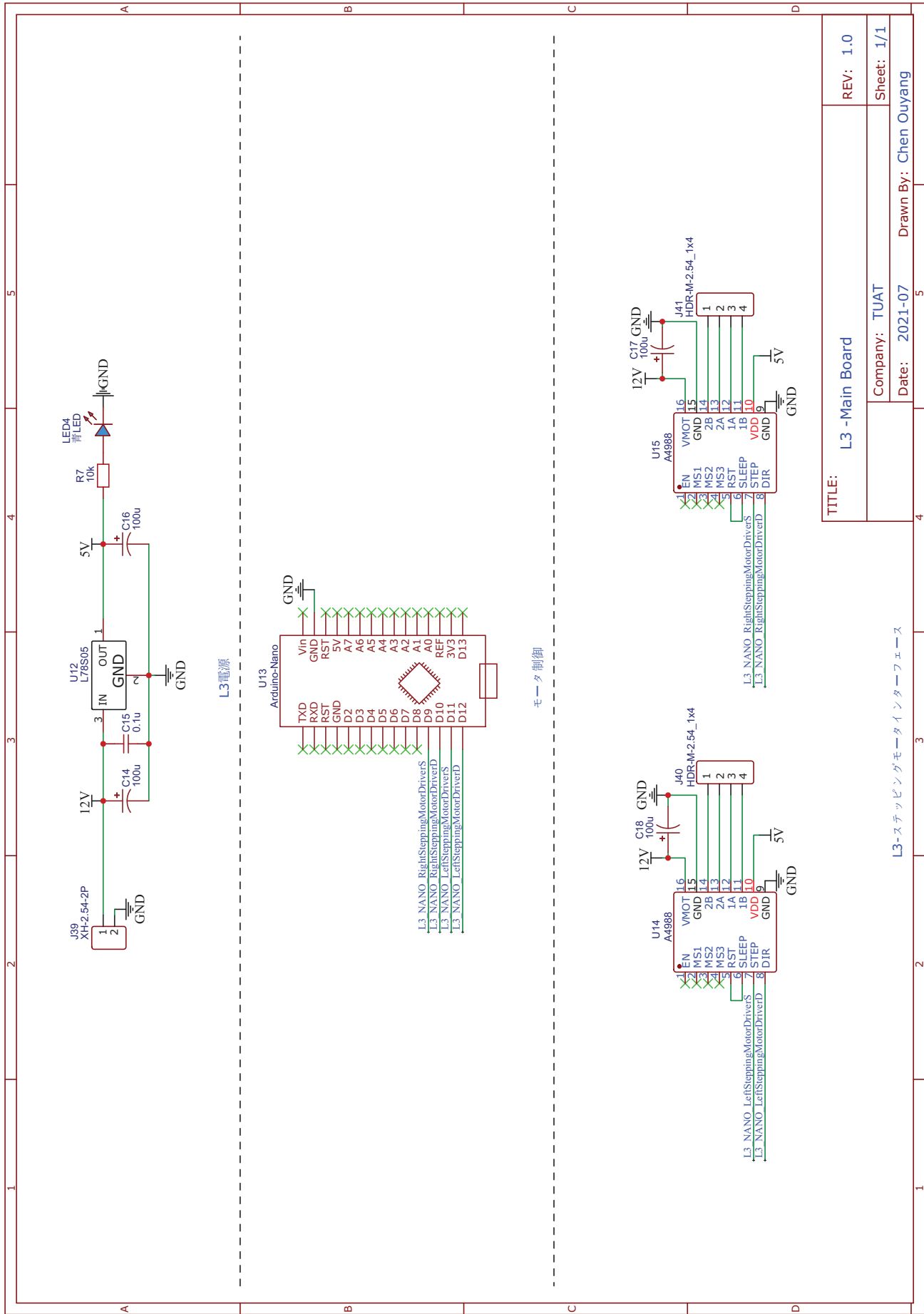


TITLE: L2-Main Board		REV: 1.0
Company: TUAT		Sheet: 1/1
Date: 2021-07		Drawn By: Chen Ouyang

## A.5 L3の回路図



TITLE: L3 -Main Board	REV: 1.0
Company: TUAT	Sheet: 1/1
Date: 2021-07	Drawn By: Chen Ouyang



TITLE: L3 -Main Board	REV: 1.0
Company: TUAT	Sheet: 1/1
Date: 2021-07	Drawn By: Chen Ouyang

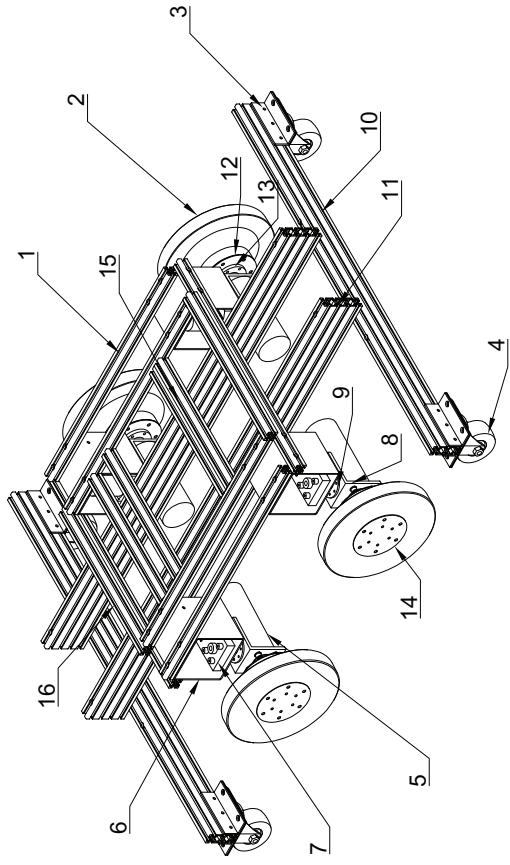
## 付録B 設計図

簡潔のため、本付録ではロボットの図面ビュー（部品リストと各モジュールの詳しい構成）を示す。

## B.1 モバイルシャーシモジュール



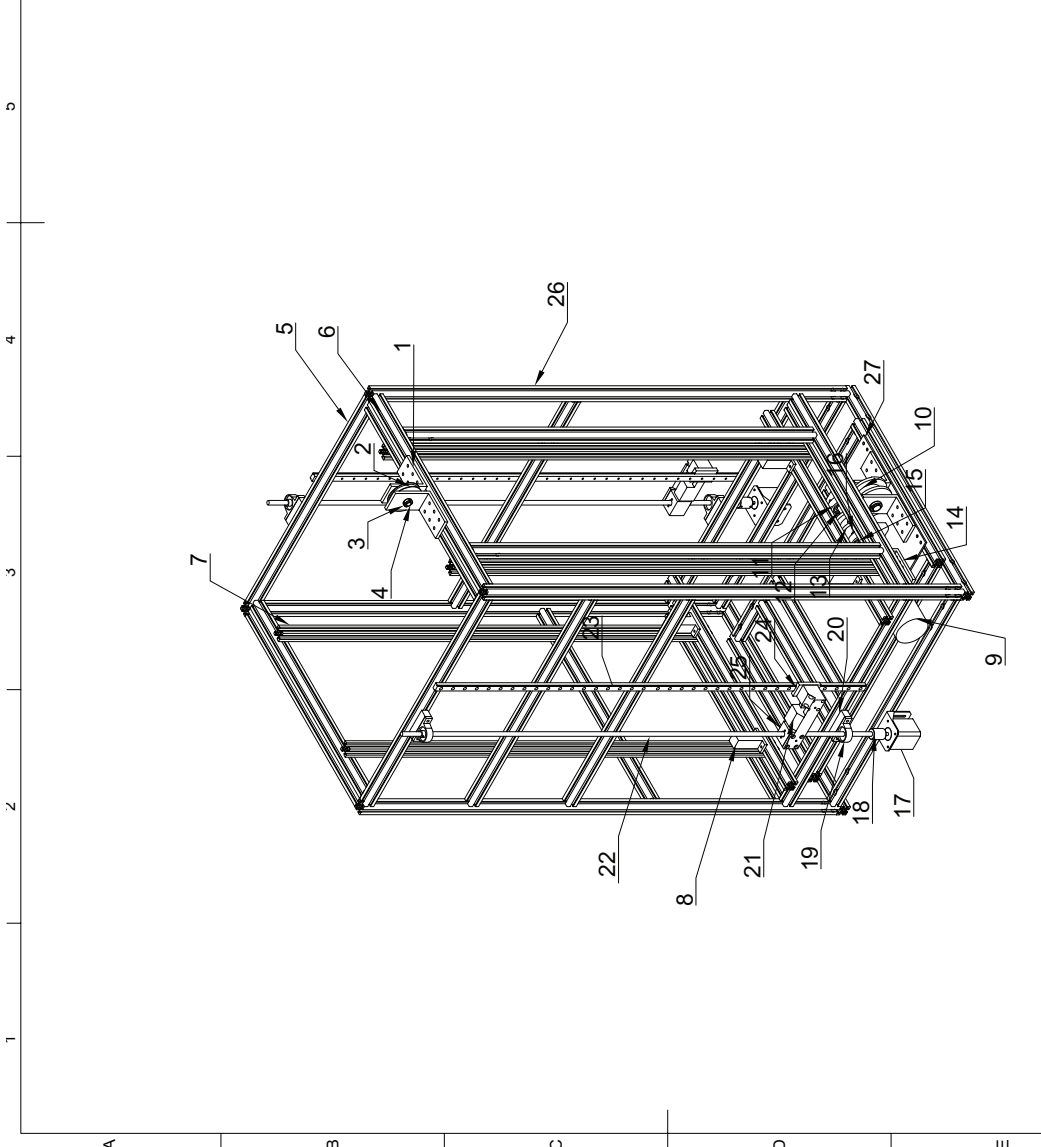
		部品表 (ネジ・ナット・ブラケット省略)			
項目	数量	部品名称	寸法 (mm)		
1	6	アルミフレーム	500x20x20	A	
2	4	ホイール	-		
3	8	アルミアングル	-		
4	4	キヤスタ	-		
5	4	直流モータ	-		
6	4	黒皮角パイプ	-		
7	4	サーボモータ	-	B	
8	4	アルミアングル	-		
9	4	dカット付フランジ	-		
10	2	アルミフレーム	830x60x20		
11	2	アルミフレーム	1000x80x20		
12	4	アルミ円板	-		
13	4	キー溝付フランジ	-	C	
14	4	アルミ円板	-		
15	4	アルミフレーム	300x20x20		
16	2	アルミフレーム	340x20x20	D	
E					
Dept.		Created by	Approved by	F	
		Chen Ouyang	2022/12/20		
Technical reference		Document type		Document status	
Title		DWG No.			
Mobile Chassis Module Parts List					
Rev.		Date of Issue		Sheet	
				1/1	



## **B.2 L1**

部品表 (ネジ・ナット・ブラケット省略)			
項目	数量	部品名称	寸法 (mm)
1	4	アルミアングル	-
2	1	滑車	-
3	2	セットカラ	-
4	1	回転軸	10x35
5	9*	アルミフレーム	500x20x20
6	9	アルミフレーム	460x20x20
7	4	V溝アルミフレーム	900x20x20
8	4	TPU緩衝材	-
9	1	直流モータ	-
10	1	滑車	-
11	2	ベアリングユニット	-
12	1	キー溝付回転軸	12x115
13	1	ローラ	-
14	1	アルミアングル	-
15	1	カップリング	-
16	2	セットカラ	-
17	2	ステッピングモータ	-
18	2	カップリング	-
19	4	ベアリングユニット	-
20	4	マット	-
21	2	カメラマウント	-
22	2	送りねじ	8x1000
23	2	リニアガイド	-
24	2	スライドブロック	-
25	2	ナットブラケット	-
26	4	アルミフレーム	1000x20x20
27	1	アルミ板	-

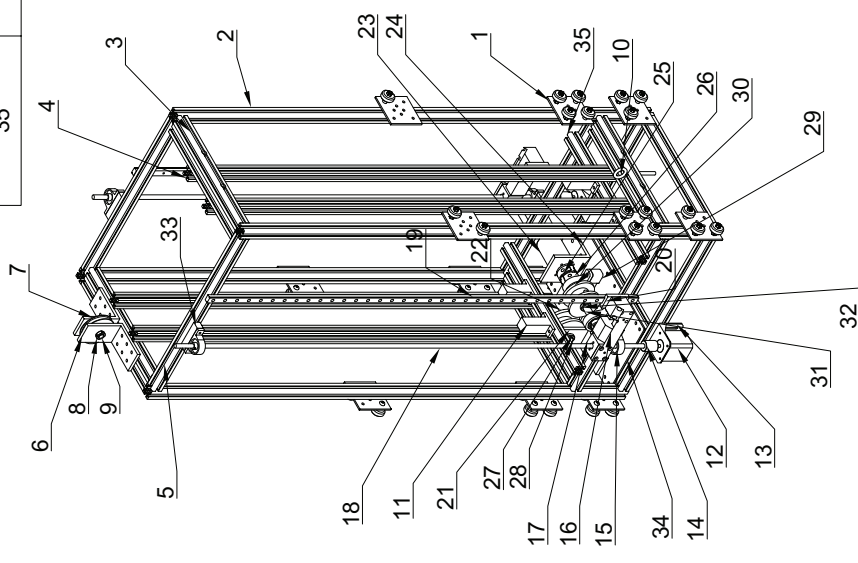
\* 実機では重量やコストを抑えるためにL1の下部をMobile Chassis Moduleの上部と合併したため、この数字は実際に使われたアルミフレームの数で、図示と一致しない。



Dept.	Created by <b>Chen Ouyang</b>	Approved by	8
Technical reference	2022/12/20	Document status	7
	Document type	DWG No.	1/1
	Title <b>Nestable Module L1 Parts List</b>		
	Rev.	Date of Issue	Sheet
			1/1

## **B.3 L2**

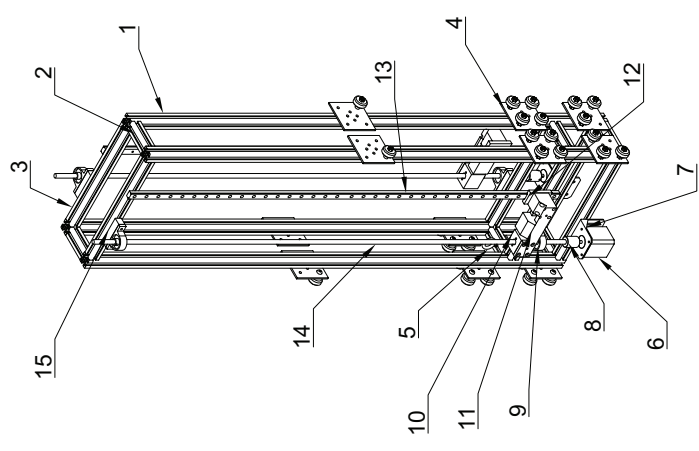
部品表(ネジ・ナット・ブラケット省略)			部品表(ネジ・ナット・ブラケット省略)				
項目	数量	部品名称	寸法 (mm)	項目	数量	部品名称	寸法 (mm)
28	1	回転軸	15x55	1	12	ガントリープレート	-
29	2	マット	-	2	4	アルミフレーム	1000x20x20
30	1	ローラ	-	3	11	アルミフレーム	260x20x20
31	1	キー溝付回転軸	12x110	4	4	V溝アルミフレーム	900x20x20
32	2	セットカラ	-	5	6	アルミフレーム	368x20x20
33	4	マット	-	6	4	アルミアングル	-
34	1	アルミ板	-	7	1	滑車	-
35	2	アルミフレーム	300x20x20	8	2	セットカラ	-
				9	1	回転軸	10x35
				10	1	アイボルト	M6
				11	4	TPU緩衝材	-
				12	2	ステッピングモータ	-
				13	2	モータ取付ブラケット	-
				14	2	カップリング	-
				15	4	ベアリングユニット	-
				16	2	カメラマウント	-
				17	2	ナットブラケット	-
				18	2	送りねじ	8x1000
				19	2	リニアガイド	-
				20	2	スライドブロック	-
				21	2	アルミアングル	-
				22	1	滑車	-
				23	1	直流モータ	-
				24	1	アルミアングル	-
				25	1	カップリング	-
				26	2	ベアリングユニット	-
				27	2	セットカラ	-



Dept.	Technical reference	Created by <b>Chen Ouyang</b>	2022/12/20	Approved by	
		Document type		Document status	
		Title	<b>Nestable Module L2 Parts List</b>		
		DWG No.			
		Rev.	Date of issue	Sheet	1/1

## **B.4 L3**

部品表(ネジ・ナット・ブラケット省略)			
項目	数量	部品名称	寸法(mm)
1	4	アルミフレーム	1000x20x20
2	6	アルミフレーム	60x20x20
3	6	アルミフレーム	236x20x20
4	6	ガントリープレート	-
5	1	アイボルト	M6
6	2	ステッピングモータ	-
7	2	モータ取付ブラケット	-
8	2	カップリング	-
9	4	ベアリングユニット	-
10	2	ナットブラケット	-
11	2	カメラマウント	-
12	2	スライドブロック	-
13	2	リニアガイド	-
14	2	送りねじ	8x1000
15	4	マット	-



Dept.	Created by <b>Chen Ouyang</b>	2022/12/20	Approved by
Technical reference	Document type		Document status
	Title	DWG No.	
	<b>Nestable Module L3 Parts List</b>		
	Rev.	Date of issue	Sheet
			<b>1/1</b>