UAV画像と深層学習によるマツ枯れ被害木の自動検出

小林　裕之・松浦　崇遠・中島　春樹

Automatic Detection of damaged trees with Pine Wilt Disease based on UAV images and Deep Learning

Hiroyuki KOBAYASHI, Takatoh MATSUURA, Haruki NAKAJIMA

富山県農林水産総合技術センター
森林研究所研究報告

No.17　令和7年3月31日　発行

Reprinted from

BULLETIN
OF
THE TOYAMA FORESTRY RESEARCH INSTITUTE

No.17　2025.3

富山森林研報　17（2025）

【論文】

UAV 画像と深層学習によるマツ枯れ被害木の自動検出

小林　裕之・松浦　崇遠・中島　春樹

Automatic Detection of damaged trees with Pine Wilt Disease based on UAV images and Deep Learning

Hiroyuki KOBAYASHI, Takatoh MATSUURA, Haruki NAKAJIMA

富山県内の 2 地区の海岸クロマツ林において，2019～2021 年に，異なる機種，異なる時期，異なる時間帯に撮影された UAV 画像から 20,200 枚のアノテーション画像を作成した。その際，被害形態は，初期，中期，末期の 3 クラスとした。まず 10%をテスト用に除外し，残り 90%を 5 等分したのち，5 分割交差検証法で YOLOv5 の small，medium，large モデルによる物体検出モデルを作成した。15 ケースの学習済みモデルのうち，学習打ち切りの判断に使用される fitness 値が最も大きかった，medium モデルの Fold 4 をベストモデルとして採用した。このモデルの mAP50 の値は 0.994 であり，既往研究よりも高い値を示した。ベストモデルをテスト用データに適用して精度検証を行ったところ，mAP50 の値は 0.994 となり，十分な精度が出ていると評価できた。2024 年 10 月 31 日に撮影した UAV 画像から，ベストモデルによるマツ枯れ被害木の自動検出を行い，別途実施した地上調査結果と比較したところ，被害木 51 本中 45 本の検出に成功し，その検出率は 88.2%となった。

1.　はじめに

松くい虫被害（以下「マツ枯れ」）は我が国最大の森林病虫害であり，マツノザイセンチュウがマツの樹体内で活動することにより，通水阻害を引き起こしてマツを衰弱・枯死させるものである（林野庁 2024）。このマツ枯れ対策には，見落としなく被害木を探し出して確実に駆除することが重要であり，地上探査と空中探査の 2 つの手法を併用することが理想的である（森林総合研究所 2022）。

最近の空中探査による研究事例としては，国内では小林・松浦（2022），Oide *et al.*（2022）が，また，国外では Zhou *et al.*（2022），Ye *et.al.*（2023）などがある。小林・松浦（2022）は，富山県の海岸クロマツ林を対象に，UAV 空撮画像の目視判読を行い，61.5%のマツ枯れ木が検出できた，としている。Oide *et al.*（2022）は，青森県のマツ林を対象に，UAV の可視光画像と機械学習を使用してマツ枯れ被害木の検出を行ったところ，精度指標のひとつである accuracy の値が 0.990～0.995 であったとしている。Zhou *et al.*（2022）は，中国遼寧省のマツ林を対象に，UAV の可視光画像から，深層学習を利用した物体検出アルゴリズムのひとつである YOLOv5（YOLO version 5）でマツ枯れ木の自動検出を行ったところ，精度指標のひとつである mAP50 の値が 0.686～0.799 であった，としている。Ye *et al.*（2023）は中国江蘇省および江西省のマツ林を対象に，YOLOv5 でマツ枯れ木の自動検出を行ったところ，mAP50 の値が 0.862 であった，としている。

小林・松浦（2022）はマツ枯れ木の検出に目視判読を利用しているが，これにはある程度の経験が必要であり，また，判読者の主観によって結果が左右されるという欠点もある。Oide *et al.*（2022）は，可視光画像から得られる R，G，B および，H，S，V バンドの画素値を使用して被害木と健全木を分類しているが，画像そのものからの直接検出はできず，また，一般的には機械学習よりも深層学習の方が優れているとも言われている。一方 Zhou *et al.*（2022）や Ye *et al.*（2023）はマツ枯れ木の検出に物体検出 AI を利用しており，この方法では解析者の主観に左右さ

れない自動検出が直接画像から可能であり，しかもその検出精度は高い。

国内を対象とした物体検出AIによるマツ枯れ検出の学術論文は存在せず，また，Zhou *et al.*（2022）やYe *et al.*（2023）は内陸のマツ林を対象としていることから，国内の海岸マツ林を対象とした物体検出AIによる研究事例はない。

そこで本研究では，Zhou *et al.*（2022），Ye *et.al.*（2023）で実績があるYOLOv5を利用して，国内の海岸マツ林におけるマツ枯れ木の自動抽出を行った結果について報告する。

2. 方法

2.1 調査地

調査対象地は，富山県下新川郡入善町目川（以下「目川」）と富山県富山市四方（以下「四方」）の2地区であり，両地区共にクロマツ（*Pinus thunbergii*）を主体とした海岸防災林である。目川については2019～2021年，四方については2020～2021年の，それぞれ初夏から晩秋にかけて，UAVによる空撮調査と地上調査を定期的に行った。林分調査を実施した時点のクロマツの状況を表-1に示す。両地区とも地上調査後に発生したマツ枯れ木の伐倒駆除処理により，クロマツの本数は年々減少した。また，四方にはクロマツ以外にニセアカシア（ハリエンジュ）等の広葉樹が存在する。

表-1 調査対象地のクロマツの林分調査結果

地区名	平均上層木高 (m)	平均胸高直径 (cm)	本数密度 (本/ha)	調査時期 (年/月)
目川	10.3	14.3	2,208	2018/6
四方	16.3	24.4	691	2020/6

2.2 使用したUAVと空撮画像

空撮に使用したUAVはいずれもDJI社の，Phantom 4 Pro（以下，「P4P」），Phantom 4 RTK（以下，「P4R」），Mavic Air（以下，「MA」）の3機種である。P4PとP4Rについては，両地区共に高度50mで自動飛行を行い，空撮画像の地上分解能は2機種共に1.3cm/pixである。MAについては目川において，バッテリーの持続時間を考慮して高度60mで自動飛行を行い，空撮画像の地上分解能は1.8cm/pixである。空撮画像からは，Metashape Pro（Agisoft）でオルソモザイク画像も作成した。なお，オルソモザイク画像の地上分解能は，オリジナルの地上分解能のままである。また，目川，四方の空撮面積はそれぞれ2.7，3.1haである。2地区で行ったすべての空撮のうち，本研究での物体検出モデルの作成に使用したものを表-2に示す。

物体検出の実施に当たっては，モデルに堅牢性を持たせるため，異なる機種，異なる地区，異なる時期と時刻，異なる画像の種類（生画像/オルソ画像）が混合するように配慮した。

表-2 物体検出に使用した空撮画像の一覧

撮影年月日	撮影開始時刻	地区名	UAV機種	アノテーション 生画像 (枚)	アノテーション オルソ画像 (枚)
2019/9/18	12:12	目川	P4P	876	0
2020/8/21	10:04	〃	〃	459	0
2020/9/15	13:39	四方	〃	387	0
2020/11/6	10:28	目川	〃	781	0
2021/7/13	9:46	〃	P4R	0	24
〃	12:50	四方	〃	0	46
2021/7/26	9:18	目川	〃	0	29
〃	11:07	四方	〃	0	32
2021/8/12	9:37	目川	〃	0	25
〃	11:23	四方	〃	0	69
2021/8/23	9:48	目川	〃	866	20
〃	13:10	四方	〃	0	47
〃	10:28	目川	MA	527	0
2021/9/7	9:27	四方	P4R	0	29
〃	11:16	目川	〃	0	30
2021/9/21	9:39	四方	〃	0	33
〃	11:24	目川	〃	0	34
2021/10/4	9:43	〃	〃	0	27
〃	12:48	四方	〃	0	42
2021/10/19	9:33	〃	〃	667	0
			計	4,563	487
			合計		5,050

P4P:Phantom 4 Pro, P4R:Phantom 4 RTK, MA:Mavic Air

2.3 画像のアノテーション

撮影した生画像と作成したオルソモザイク画像から，pictcutter（まっつん工房）によって被害樹冠を含む5,050枚（表-2）の画像（640×640画素）を切り出し，labelImg（Tzu Ta）によって，被害樹冠を囲む，バウンディングボックスと呼ばれる矩形領域の作成と被害形態のクラス分け（アノテーション）を行った。被害形態は，Zhou *et al.*（2022）にならい，針葉が変色し始めた初期（”yellow”），針葉が赤く枯れた中期（”red”），萎凋した針葉と枝の塊となった末期（”branch”），の3クラスとした。各クラスの被害形態のラベル名とインスタンス数（バウンディングボックスの数）を表-3に，また，アノテーションの例を図-1に示す。

表-3　被害形態のラベル名とインスタンス数

被害段階	被害形態	ラベル名	インスタンス数
初期	樹冠が黄緑色～黄色(枯れ始め)	yellow	2,217
中期	樹冠が赤色(枯れ)	red	2,721
末期	枯死後の枯葉と枝の塊	branch	4,754
		計	9,692

図-1　アノテーションの例

2.4 YOLOv5

AI（人工知能）の一分野に機械学習があり，機械学習の一手法に深層学習（ディープラーニング）がある（我妻 2022）。画像の「どこ」に「なに」が「何%の確信度」で存在するかという情報を取得する物体検出は，この深層学習を利用している（山口・松田 2019）。物体検出には，R-CNN，Fast R CNN，Faster R-CNN，YOLO（You Only Look Once），SSD（Single Shot Multibox Detector）などの手法があるが，これらのうち，高速，高精度で一番早いアルゴリズムがYOLOであり（川島 2019），本研究ではZhou *et al.*（2022），Ye *et.al.*（2023）が使用した，YOLO version 5（YOLOv5）を使用した。

まずローカルPC上にYOLOv5の動作環境を構築した。使用したPCの主な仕様と使用したソフトウェアを表-4，5にそれぞれ示す。環境設定は，NVIDIA社製GPU用のCUDA ToolkitとcuDNN（いずれもNVIDIA）をインストールしたのち，Anaconda（ANACONDA）をインストールし，その仮想環境上でPyTorch（PYTORCH FOUNDATION）とYOLOv5（Ultralytics）をインストールする，という手順で行った（Nukui 2022）。

表-4　使用したPCの主な仕様

	CPU	メモリ	GPU（グラフィックボード）
PC1	Corei5-14600K	64GB	4070 Ti SUPER（16GB）
PC2	Ryzen 9 5950X	128GB	4070 Ti SUPER（16GB）
PC3	Corei5-13600K	64GB	3070（8GB）
PC4	Corei7-14700K	128GB	4060 Ti（16GB）
PC5	Ryzen 9 5950X	64GB	4060 Ti（16GB）
PC6	Ryzen 9 3900X	64GB	4070（12GB）

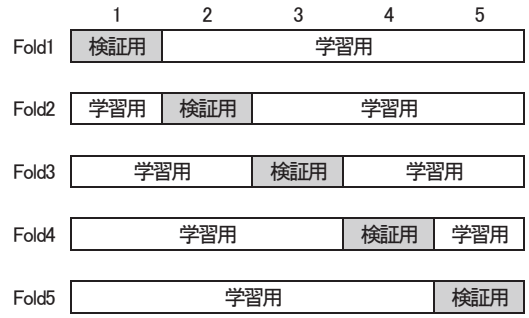
GPUはいずれもNVIDIA社のGEFORCE RTXシリーズ
OSはいずれもMicrosoft社のWindows 11 Pro

表-5　使用したソフトウェア

区分	名称
Python開発環境	Anaconda
機械学習ライブラリ	PyTorch
GPUプログラム開発環境	CUDA Toolkit
深層ニューラルネットワーク用ライブラリ	cuDNN
物体検出アルゴリズム	YOLOv5

最高のトレーニング結果を得るためには1クラスあたり1,500枚以上の画像と，1クラスあたり10,000個のインスタンス数（バウンディングボックスの数）が推奨されている（Ultralytics 2024a）。本研究でアノテーション作業を行った画像数は5,050枚であり（表-2），推奨値（3クラス×1,500枚＝4,500枚）を満たしていたが，インスタンス数は3クラス合計で9,692個（表-3）と推奨値（3クラス×10,000＝30,000インスタンス）を満たしていなかったことから，データ数を増やす手法である，データ拡張（水増し）を行った。データ拡張は，アノテーションデータのバウンディングボックスを上下，左右，上下左右に反転するPythonプログラム（Koga 2021）により実施した。この処理により画像数とインスタンス数はそれぞれ4倍の20,200枚，38,768インスタンスとなった。

物体検出モデルの精度を検証するために，本研究では20,200枚のアノテーション済み画像のうちの10%（2,020枚）をまずテスト用として除外したのち，残りの18,180枚を5等分し（Fold 1～Fold 5，各3,636枚），5分割交差検証法によってモデルの作成と精度評価を行い，テストデータによる性能評価を行った。5分割交差検証のイメージを図-2に示す。なお，データの分割は，R（R Development Core Team）のスクリプトによるランダム選択で行った。



各 Fold とも，学習用:14,544 枚，検証用:3,636 枚

図-2　5 分割交差検証のイメージ

YOLOv5 には，主として 3 つの Python プログラムがある。それらは，train.py，detect.py，ならびに val.py である。train.py は学習および検証用データを使用して学習と検証をくり返しながら学習済みモデルを作成し，detect.py はそのモデルを利用して，アノテーションがなされていない未知の画像から物体検出を行い，val.py は train.py では使用していないアノテーション済みデータから学習済みモデルを使用してその検出精度を出力するものである。

YOLOv5 は，検出精度と演算負荷に応じて，small (s)，medium (m)，large (l)，extra-large (x) の 4 つのモデルがあるが，x モデルの計算処理には膨大な計算時間が必要であったため，本研究では，s，m，l の 3 つのモデルと 5 つの Fold で train.py による学習を行い，(15 ケース)，検出精度が最も高いケースをベストモデルとして採用し，テスト用データによる性能評価も行った。

train.py の実行時には，batch-size（学習用画像をグループ分けする際の 1 グループあたりの枚数，規定値は 16）と epochs（学習用画像全体の最大繰り返し学習回数，規定値は 300）を指定する必要がある。batch-size は 2 のべき乗の値が使われることが多く (Sasaki 2020)，大きいほど学習が速く進行するが，GPU のメモリサイズを超えると学習速度が極端に低下することから，6 台の PC の GPU のメモリサイズ（表-4）とモデルに応じて，64，32，16，8 を指定した（表-6）。また，epochs は試行錯誤の結果，一律で 5,000 を指定し，転移学習も行った。

採用したベストモデルを利用し，2024 年時点でもマツ枯れ被害が発生している四方を対象に，UAV 空撮画像から被害木の自動抽出を行い，現地調査結果との比較を行い，モデルの精度を検証した。

3. 結果と考察

3.1 学習結果とテストデータでの評価

学習結果の一覧を表-6 に示す。train.py プログラムは 1 epoch ごとに fitness という精度指標を計算し，直近 100 epoch でその fitness 値に向上がなかったときに学習を打ち切り，fitness 値最大のときの学習済みモデルを best.pt ファイルとして記録し (Ultralytics 2024b)，この best.pt を使用して detect.py や val.py を実行する。

fitness 値が最大となったときの epoch (best_epoch) は 912 (l モデルの Fold 5) ～ 3,858 (s モデルの Fold 3) であった。また，学習が収束するまでの所要時間は，41.6 (s モデルの Fold 1) ～251.8h (m モデルの Fold 5) であり，平均所要時間は 130.5h (5.44 日) であった。

精度指標のひとつである mAP50 の値は，0.989 (l モデルの Fold 1) ～0.994 (s モデルの Fold 2 ほか) となり，Zhou *et al.* (2022)，の 0.686 ～0.799 や Ye *et.al.* (2023) の 0.862 を上回った。また，本研究の学習用画像の一部を使用した予備研究 (Kobayashi 2023) での 0.800～0.824 も上回った。Zhou *et al.* (2022) はさらに，独自に改良した YOLOv5 でも検出を行い，その mAP50 が 0.802～0.885 であったとしているが，本研究ではそれらをも上回り，高精度な検出精度が得られた。

全 15 ケースのうち，fitness 値が最も大きかったのは m モデルの Fold 4 (0.975) であり，本研究ではこれをベストモデルとして選定した。ベストモデルのクラス別検出精度を表-7 に示す。この表からは，3 つのクラス間で 4 つの精度指標 (precision，recall，mAP50，mAP50-95) にほとんど差がなく，各クラス満遍なく高精度な検出結果が得られたと考えられる。

表-7　ベストモデルのクラス別検出精度

クラス	画像枚数	インスタンス数	precision	recall	mAP 50	mAP 50-95
yellow	3,636	1,549	0.992	0.993	0.995	0.977
red	3,636	1,913	0.991	0.990	0.995	0.975
branch	3,636	3,441	0.989	0.994	0.993	0.967
総合	3,636	6,903	0.991	0.992	0.994	0.973

ベストモデルの YOLOv5 の出力グラフを図-3

に，また，アノテーション画像（学習用）と検出画像（検証用）の比較例を図-4 に示す。

図-3 の(1)～(3)は，epoch（繰返し回数）の増加に伴って，学習用データの誤差が最初は大きく，途中からは少しずつ減少していくという，アルファベットの「L」字に近い形状になっており，誤差が順調に減っていったことがわかる。(4)，(5)，(9)，(10)は精度指標であり，最初は大きく，途中からは少しずつ増加しており，学習打ち切りまで精度向上が続いたことがわかる。(6)，(7)，(8)は，検証用データの誤差を示しており，これらも学習用データのそれらと同様に，「L」字型の形状になっており，誤差が順調に減っていったことがわかる。特に(7)の obj_loss は，検証用画像（Fold 1～5 のうちのひとつ：3,636 枚）の検出前のアノテーション画像内のバウンディングボックスが，モデルによる自動検出後には存在しなかったり（未検出），検出前には存在しないバウンディングボックスが検出後には存在したり（誤検出）する場合に値が大きくなる誤差であるが，これも順調に減少していることがわかる。学習用データの傾向に沿うように学習させた結果，学習用データにはよい精度を出すが，検証用データに対しては同様の精度を出せないことを過学習と呼び，この過学習が生じた場合には(7)のグラフが，最初は減少したのち，途中から増加しはじめて（Ultralytics 2024c），カタカナの「レ」字に近い形状になるが，本研究では「L」字型になっているので，過学習は発生しなかったと考えられる。

図-4 からは，バウンディングボックスの位置や被害形態（yellow，red，branch）がアノテーション画像と検出結果画像で一致しており，検出の信頼度も 0.9～1.0 と非常に高いので，検出精度の高いモデルが構築できたと考えられる。

予め確保しておいた 2,020 枚のテスト用アノテーション画像に対して，ベストモデルの best.pt を使用した val.py の結果を表-8 に示す。全ての精度指標とも高い値となっており，十分な精度が出ていると評価された。

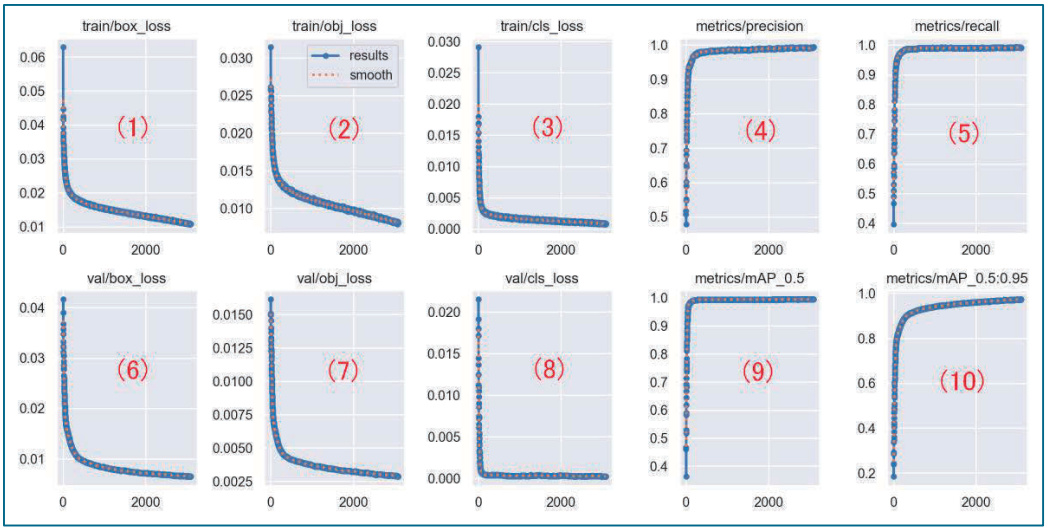
表-8　テストデータでの精度検証結果

モデル	Fold	precision	recall	mAP 50	mAP 50-95	fitness
m	4	0.991	0.992	0.994	0.975	0.977

表-6　学習結果の一覧

モデル	Fold	pc	batch_size	epochs	stop at	所用時間 (h)	best_fitness	best_epoch	precision	recall	mAP 50	mAP 50-95
s	1	PC1	64	5000	1824	41.6	0.903	1724	0.977	0.982	0.990	0.893
s	2	PC2	64	5000	3130	87.4	0.933	3030	0.990	0.991	0.994	0.926
s	3	PC3	32	5000	3958	145.0	0.937	3858	0.988	0.985	0.992	0.931
s	4	PC4	64	5000	2609	110.3	0.926	2509	0.987	0.989	0.993	0.918
s	5	PC5	64	5000	2643	117.5	0.921	2543	0.981	0.981	0.991	0.913
m	1	PC1	32	5000	2296	89.3	0.958	2196	0.985	0.985	0.990	0.954
m	2	PC2	32	5000	2205	104.6	0.964	2105	0.991	0.993	0.994	0.961
m	3	PC3	16	5000	2235	135.3	0.948	2135	0.985	0.985	0.991	0.943
m	4	PC4	32	5000	3090	233.2	0.975	2990	0.991	0.992	0.994	0.973
m	5	PC5	32	5000	3147	251.8	0.968	3047	0.988	0.983	0.991	0.965
l	1	PC1	16	5000	1489	91.2	0.958	1389	0.979	0.985	0.989	0.954
l	2	PC2	16	5000	2047	143.6	0.972	1947	0.988	0.992	0.994	0.969
l	3	PC1	16	5000	1776	108.1	0.965	1676	0.985	0.987	0.992	0.961
l	4	PC3	8	5000	1995	209.3	0.960	1895	0.986	0.988	0.993	0.956
l	5	PC6	16	5000	1012	88.9	0.956	912	0.984	0.984	0.993	0.952

pc：使用した PC（表-4 参照）
batch_size：学習用画像をグループ分けする際の 1 グループあたりの枚数
epochs：学習用画像全体の最大の繰り返し学習回数
stop at：直近 100 epoch で fitness 値（=mAP50×0.1＋mAP50-95×0.9）に向上がなく，計算が終了したときの epoch
best_fitness：最大の fitness 値
best_ecpch：fitness 値最大のときの epoch
precision：検出結果の総数のうち，実際に被害樹冠であった数の割合
recall：実際の被害樹冠の総数のうち，被害樹冠として検出された数の割合
mAP50：正解位置との IOU（BOX 同士の重なり比率）が 0.5 以上の検出結果を正解とした場合の，全クラスに対する AP（物体検出の精度）の平均値
mAP50-95：IOU を 0.5～0.95 の範囲で 0.05 刻みに変化させたときの AP の平均



(1) 学習用データの検出位置の正解値と予測値との誤差
(2) 学習用データの検出範囲内の物体の有無の正解値と予測値との誤差
(3) 学習用データのクラス分けの正解値と予測値との誤差
(4) 検出結果の中で正しく検出された割合
(5) 正解の総数に対して正しく検出された割合
(6) 検証用データの検出位置の正解値と予測値との誤差
(7) 検証用データの検出範囲内の物体の有無の正解値と予測値との誤差
(8) 検証用データのクラス分けの正解値と予測値との誤差
(9) 正解位置との IOU(BOX 同士の重なり比率)が 0.5 以上の検出結果を正解とした場合の全クラスに対する AP(物体検出の精度)の平均
(10) IOU を 0.5～0.95 の範囲で 0.05 刻みに変化させたときの AP の平均

(横軸:epoch)

図-3　ベストモデルの YOLOv5 の出力グラフ（m モデルの Fold 4）

（アノテーション画像）　　（検出結果画像）

(yellow, red, branch は 3 クラスの被害形態, 検出結果の数値は信頼度(0＜信頼度≦1))

図-4　ベストモデルのアノテーション画像と検出画像の比較例（m モデルの Fold 4）

3.2 四方の 2024 年画像からの検出

2024 年 10 月 31 日に，依然としてマツ枯れ被害が発生している四方において P4R による自動空撮を行った。空撮画像から作成したオルソモザイク画像を分割したのち，ベストモデルの best.pt を使用した自動検出を detect.py により実行した。また，空撮前に富山農林振興センターが実施した現地調査によって被害木としてマークされていたクロマツを現地で確認し，ハンディ GPS によってその位置情報を記録した。

現地調査で確認された被害木のうち，空撮範囲内に存在したものは 51 本であった。それら被害木の位置と検出されたバウンディングボックスの位置との関係を表-9 に示す。ボックス内に存在した被害木は 25 本，ボックス近傍に存在した被害木は 20 本，ボックス内とその近傍に存在しなかった被害木（未検出木）は 6 本であった。ボックス内とその近傍の被害木の一部と，未検出木のすべてを図-5 に示す。図中(b)の y25，(c)の y26，(e)の y36，y37，(f)の y48 などはボックスのすぐ近くに存在しており，位置決めに使用したハンディ GPS の測位精度を考慮すれば，検出に成功したと考えられる。

一方，図中(g)の y09 はポイント北西側に黄緑色の樹冠が見えるにもかかわらず，”yellow”として検出されなかったが，その理由は不明である。(h)の y11 については，ポイント南側に枯死樹冠が見えるが，林床の茶色と区分ができず，検出できなかったのではないかと考えられる。(i)の y12 については，ポイント東側に枯死樹冠が見えるが，樹冠が小さめで検出できなかったのではと考えられる。(j)の y 22 についても樹冠が小さめであり，検出できなかったのかもしれない。(k)の y34 については，現地調査では枯死樹冠が確認できたが，健全木の樹冠の下になっており，写真には写らずに検出できなかったと考えられる。(l)の y39 については，ポイント南西側に枯死樹冠が見えるが林床の茶色と区別できずに検出できなかったのではと考えられる。今回の空撮日は 10 月 31 日であり，既に下草が枯れて赤茶色に変色していたが，空撮日がもう少し早ければ，y11 や y39 の未検出は防ぐことができたと推察される。

被害木 51 本のうち，ボックス内とボックス近傍の被害木は 45 本であり，これらを検出成功とみなせば，本研究での検出率は 88.2%（51 本中 45 本）となり，目視判読による小林・松浦（2022）の 61.5%を上回る結果となった。

今回作成したオルソモザイク画像には，マツ林北側の草地や砂浜が写っており，そこでは多数のバウンディングボックスが“branch”として作成されていた（誤検出）。しかしながら，それらの誤検出はそもそもマツ林外で発生したものであり，現地調査時に調査対象から除外すればよいと考えられる。また，このような誤検出を減らすためには背景画像（対象物のない画像）をデータセットに追加する方法があるが（Ultralytics 2024a），これについては今後の課題としたい。

なお，マツ林内においては，落葉後のニセアカシアの樹冠が”branch”として検出されていた。ニセアカシアの誤検出の例を図-6 に示す。このような誤検出については，ニセアカシアの落葉前に空撮を行えば回避できたであろうと考えらえる。

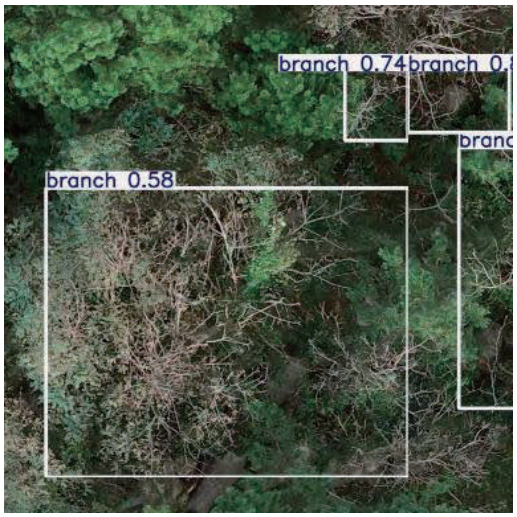


図-6　誤検出された落葉後のニセアカシア

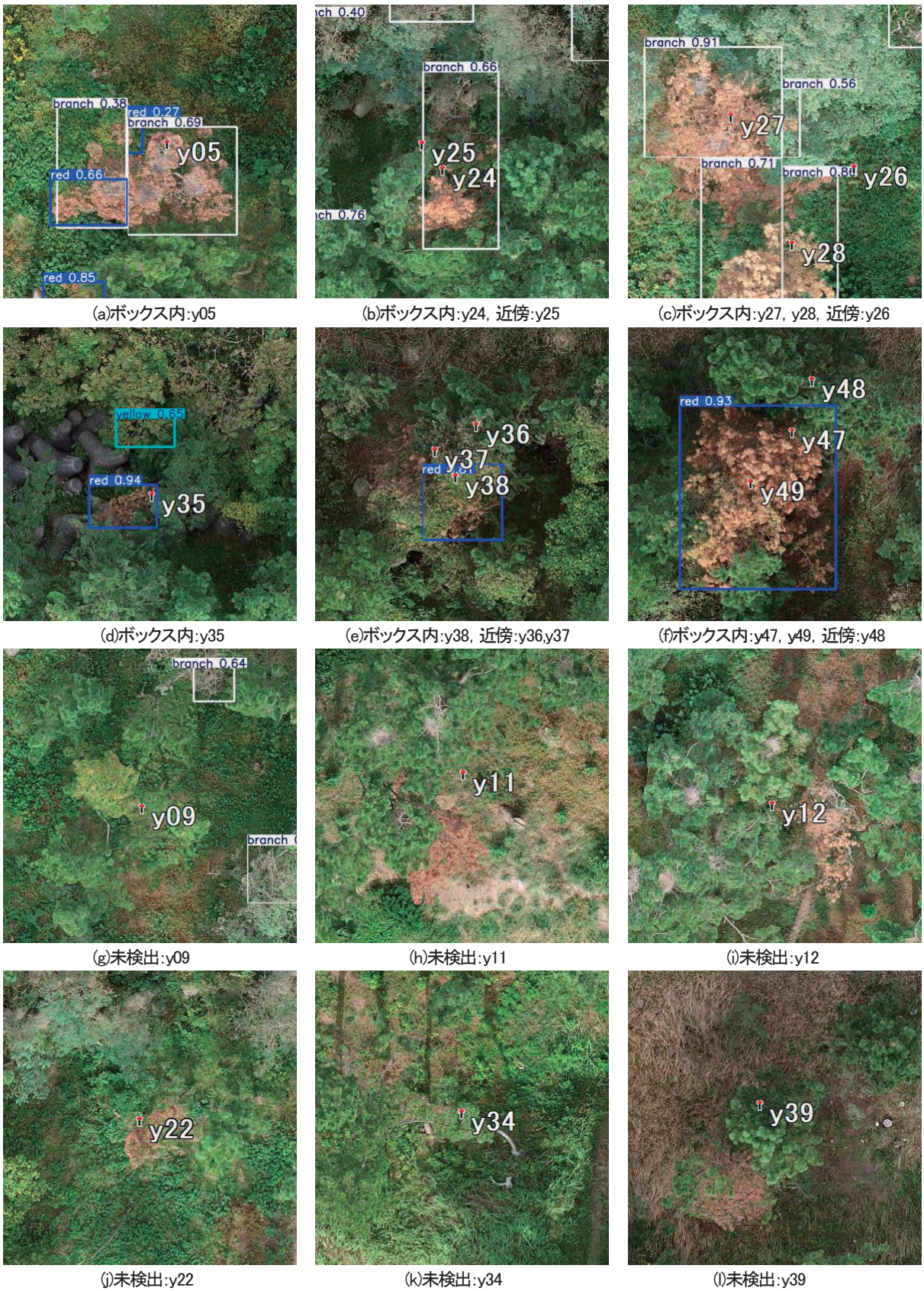
林内での未検出や誤検出を防ぐためには下草の枯れや広葉樹の落葉前に空撮を行うことが有効であると考えられるが，撮影時期を早めると，遅い時期に枯れるマツを見逃す恐れもある。これを回避するには，例えば 9 月と 10 月の 2 回空撮を行うなど，複数回の撮影が有効であると考えられる。

本研究で得られた知見等に基づき，ドローン空撮，被害木の検出から現地調査までの作業の進め方を記述した手順書（小林 2025）を作成した。

表-9　自動検出結果と現地調査結果との比較

被害木ID	バウンディングボックス内	バウンディングボックス近傍	近傍にボックスなし	DBH (cm)	備考
y01		1			
y02		1			
y03	1				
y04		1			
y05	1				
y06	1				
y07		1			
y08		1			
y09			1	19.8	ポイント北西側に黄緑色の樹冠が見えるが，"yellow"でも検出なし
y10		1			
y11			1	39.0	ポイント南側に枯死樹冠が見えるが，林床の茶色と区別ができず未検出か
y12			1	36.8	ポイント東側に枯死樹冠が見えるが，小さめなので未検出か
y13	1				
y14	1				
y15	1				
y16		1			
y17	1				
y18	1				
y19	1				
y20	1				
y21	1				
y22			1	14.9	樹冠が小さ過ぎて未検出か
y23	1				
y24	1				
y25		1			
y26		1			
y27	1				
y28	1				
y29	1				
y30	1				
y31	1				
y32	1				
y33	1				
y34			1	22.0	健全木の樹冠下に隠れて写真に写らなかったか
y35	1				
y36		1			
y37		1			
y38		1			
y39			1	23.0	ポイント南西側に枯死樹冠が見えるが，林床の茶色と区別できず未検出か
y40		1			
y41		1			
y42		1			
y43		1			
y44		1			
y45		1			
y46		1			
y47	1				
y48		1			
y49	1				
y50	1				
y51	1				
計	25	20	6		

網掛けは検出できなかったもの（未検出）



(a)ボックス内:y05　(b)ボックス内:y24，近傍:y25　(c)ボックス内:y27，y28，近傍:y26

(d)ボックス内:y35　(e)ボックス内:y38，近傍:y36,y37　(f)ボックス内:y47，y49，近傍:y48

(g)未検出:y09　(h)未検出:y11　(i)未検出:y12

(j)未検出:y22　(k)未検出:y34　(l)未検出:y39

（図中の赤いピンは現地調査時にハンディ GPS で取得した被害樹冠のウェイポイント）

図-5　四方地区 2024 年空撮画像からの自動検出結果と現地調査結果の比較

謝辞

本研究の遂行にあたり，目川および四方地区のマツ林に関する情報を提供していただいた，新川および富山農林振興センターの職員の方々に感謝します。また，YOLOv5 のローカル PC での動作環境の構築から検出モデルの作成に関して指導，助言をいただいた，富山県農林水産総合技術センター園芸研究所果樹研究センター主任研究員の杉山洋行氏に感謝申し上げます。

引用文献

我妻幸長（2022）PyTorch で作る！深層学習モデル・AI アプリ開発入門．翔泳社

川島 賢（2019）今すぐ試したい！機械学習・深層学習（ディープラーニング）画像認識プログラムレシピ．秀和システム

小林裕之・松浦崇遠（2022）UAV による海岸林のマツ枯れ被害の時系列観測．日林誌 104：99–105

Kobayashi H（2023）UAV 画像と深層学習によるマツ枯れ被害木の検出．第 135 回日本森林学会大会講演要旨集．https://www.forestry.jp/content/images/2024/04/%E6%A3%AE%E6%9E%97%E5%AD%A6%E4%BC%9A%E8%AC%9B%E6%BC%94%E8%A6%81%E6%97%A8%E9%9B%86135_%E5%86%8A%E5%AD%90%E7%89%88.pdf（最終アクセス日：2024 年 12 月 22 日）

小林裕之（2025）ドローンと AI を活用したマツ枯れ被害調査の作業手順．富山森林研報 17（印刷中）

Koga T（2021）【YOLO】labelImg でアノテーション済みデータを増殖（data augmentation）．https://qiita.com/garcoo/items/2ab8972ce304ffc8d225（最終アクセス日：2024 年 12 月 11 日）

Nukui T（2022），【YOLOv5】スナックエンドウの収穫に物体検出をつかってみる【独自データ】．https://farml1.com/yolov5/（最終アクセス日：2024 年 12 月 10 日）

Oide A H, Nagasaka Y, Tanaka K（2022）Performance of machine learning algorithms for detecting pine wilt disease infection using visible color imagery by UAV remote sensing. Remote Sensing Applications: Society and Environment 2022 (28) https://doi.org/10.1016/j.rsase.2022.100869（最終アクセス日：2025 年 2 月 12 日）

林野庁（2024）松くい虫被害対策について．https://www.rinya.maff.go.jp/j/hogo/higai/attach/pdf/matukui_R5-11.pdf（最終アクセス日：2024 年 12 月 9 日）．

Sasaki K（2020）機械学習／ディープラーニングにおけるバッチサイズ，イテレーション数，エポック数の決め方．https://qiita.com/kenta1984/items/bad75a37d552510e4682（最終アクセス日：2024 年 12 月 16 日）

森林総合研究所（2022）マツ材線虫病にどう対処するか－防除対策の考え方と実践－．https://www.ffpri.affrc.go.jp/pubs/chukiseika/documents/5th-chuukiseika11.pdf（最終アクセス日：2024 年 12 月 9 日）．

Ultralytics（2024a）最高のトレーニング結果を得るためのヒント．https://docs.ultralytics.com/ja/yolov5/tutorials/tips_for_best_training_results/（最終アクセス日：2024 年 12 月 11 日）

Ultralytics（2024b）フィットネスの定義．https://docs.ultralytics.com/ja/yolov5/tutorials/hyperparameter_evolution/#2-define-fitness（最終アクセス日：2024 年 12 月 16 日）

Ultralytics（2024c）オーバーフィット．https://www.ultralytics.com/ja/glossary/overfitting（最終アクセス日：2024 年 12 月 22 日）

山口達輝・松田洋之（2019）図解即戦力機械学習＆ディープラーニングのしくみと技術がこれ 1 冊でしっかりわかる教科書．技術評論社

Ye X, Pan J, Liu G, Shao F（2023）Exploring the close-range detection of UAV-based images on Pine Wilt Disease by an improved Deep Learning method. Plant Phenomics 2023. https://doi.org/10.34133/plantphenomics.0129（最終アクセス日：2024 年 12 月 9 日）

Zhou Y, Liu W, Bi H, Chen R, Zong S, Luo Y（2022）A detection method for individual infected pine trees with Pine Wilt Disease based on Deep Learning. Forests 2022 (13), 1880. https://doi.org/10.3390/f13111880（最終アクセス日：2024 年 12 月 9 日）

Summary

We created 20,200 annotated images from UAV images of coastal forests of Japanese black pine in

two districts of Toyama Prefecture. These images were taken by using different machine models, in different seasons, and at different times during the period 2019 to 2021. Pine wilt damage phases were classified as early, middle, and late. First, we excluded 10% of the images for testing purposes, and the remaining 90% were equally divided into five. We then used the 5-fold cross-validation method to create an object detection model by using small, medium, and large models of YOLOv5. Among the 15 learned models, we adopted fold 4 of the medium model, in which the fitness value used for the decision of learning discontinuation was the largest, as the best model. The mAP50 value of this model was 0.994, which was higher than that of any preceding studies. Precision verification of the best model by using the data for testing revealed that the mAP50 value was 0.994, meaning that sufficient precision was obtained. The results of automatic detection of pine trees affected by pine wilt disease by using the best model derived from the UAV images taken on 31 October 2024 and the results of an on-site survey conducted separately were compared. The comparison showed that 45 of the 51 affected trees were successfully detected by automatic detection; that is, the detection rate was 88.2%.