

AI を利用した風車点検用ドローンの自律飛行機能の開発

Development of autonomous flight function of the drone for inspecting wind turbines using AI

機械加工ロボット科 三浦勝吏 太田悟 塚本遊
応募企業 株式会社福島三技協

AI 技術を風車点検用ドローンに応用するため、リアルタイムで物体検出が可能な YOLO を用いてブレードを検出し、先端まで追従飛行する手法を検討した。ブレード認識の学習モデルを構築し、YOLO とドローンの動作を ROS で制御することによって、ブレードに追従飛行することができた。これにより、風車点検用ドローンの自律飛行の可能性を示すことができた。

Key words: 風力発電、ドローン、AI、YOLO、ROS、物体検出、物体追従飛行

表 1 開発環境

ドローン	DJI社製Tello Edu
ドローン搭載カメラ	HD720p/30fps
フレームワーク	ROS-melodic
OS	Ubuntu18.0
GPU	GeForce RTX2080Ti

1. 緒言

提案企業である株式会社福島三技協では、風力発電用風車の雷対策用接地線の断線有無を検査する点検ドローンの開発に取り組んでいる。これまでの断線検査ではロープワークやクレーンを使用し、人がブレードまで近づいて点検してきたが、危険が伴う作業となる、点検費用が高額になるなどの課題があった。そこで、提案企業では、ドローンを活用して点検する手法の開発を進めている。点検はドローンに取り付けているセンサと風車のブレード先端を接触させて行う。手動操縦での点検には成功したが、ドローンの精細な操作が必要で、訓練に多くの時間を要するという課題が新たに生じた。そのため、誰もが容易に点検できるように、カメラ画像からブレードを検出し、ブレード先端までドローンが自動で飛行するシステムを開発したいとの要望があった。

そこで、本研究では、リアルタイムで物体検出が可能な YOLO^{1),2)} と小型のドローンを用いてブレードを検出し、ブレードまで追従飛行するシステムを構築し模擬実験を行うことで、実際に点検ドローンの自律飛行が可能か検討した。

2. 実験及び結果

2. 1. 環境構築

2. 1. 1. 使用機器とソフトウェア

ドローンと制御用 PC を Wi-Fi (IEEE 802.11n) で接続し、映像や制御信号などのデータの送受信をすることで物体追従飛行が可能な環境を構築した。ドローンは DJI 社製 Tello Edu³⁾ を用いた。

また、制御システムを構築する際、ロボットシステム開発用のオープンソフトウェアフレームワークである Robot Operation System⁴⁾ (以下、ROS) を使用した。開発環境を表 1 に示す。

さらに、YOLO を用いて、リアルタイムにブレードを

検出する学習モデルを構築し、上記開発環境下でブレードを認識し、接近していく自律飛行機能を開発した。

2. 2. YOLO による物体認識

2. 2. 1. 学習モデルの構築

YOLO は、様々なフレームワークで実装されたものが提供されており、本研究ではハイテクプラザで動作実績のある Darknet⁵⁾ を利用した。学習は YOLO-v3 の学習モデルをもとにクラス数や入力データサイズを変更して行った。

実際にドローンで点検する際は、ドローンにカメラを上向きに取り付けることを想定しているため、学習には、下から風車を見上げた映像が必要である。そこでカメラを上向きに取り付けたドローンを、風車の真下からブレードに接近させ、ブレードと支柱の画像をそれぞれ 500 枚収集した。画像収集には国立研究開発法人産業技術総合研究所福島再生可能エネルギー研究所 (以下、FREA) に設置されている高さ約 40[m] の株式会社駒井ハルテック製の風車を使用した。学習用の画像データの例を図 1 に示す。

学習モデルは、風車のブレード (blade) と支柱 (prop) の 2 クラスを検出できるように構築した。YOLO は検出物をラベル付きの矩形 (BoundingBox) で囲むことで検出物が画像中のどこに位置するかを示す。画像中のどの領域が認識してほしい物体であるか手作業で指定するアノテーション作業には BBox-Label-Tool⁶⁾ を使用した。アノテーション後のデータを使い学習した。学習に用いた際の計算機の仕様を表 2 に示す。



図1 学習用画像データ例

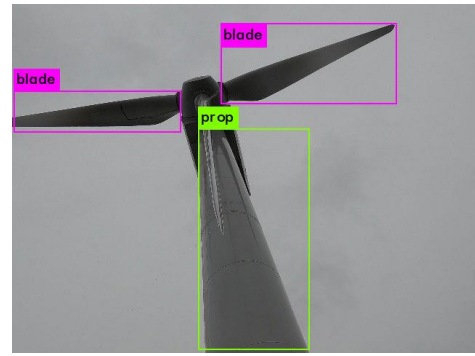


図2 検出結果

表2 学習に用いた計算機の仕様

OS	Ubuntu18.04LTS
メモリ	32GB
CPU	Core i9 10900X
GPU	Geforce RTX2080Ti
GPGPUプラットフォーム	CUDA V10.1
学習モデル	YOLO-v3
フレームワーク	Darknet

2. 2. 2. ブレードの検出結果

学習に使用していない画像を用いて風車を検出したところ、ブレード及び支柱を検出していることが確認できた。検出結果の例を図2に示す。この学習済みのモデルを用いることでカメラ映像からリアルタイムでブレード及び支柱を検出することが可能となった。表1に示す開発環境では約 30[fps]で検出可能であった。

2. 3. 物体追従飛行

2. 3. 1. ROSによる追従飛行機能の開発

ROSは、「ノード」と呼ばれるプログラムを複数実行し、ノード間でデータのやり取りをすることで1つのシステムを構成することができる。ノード間のデータのやり取りは「トピック」に値を入れ込む形でデータを送受信している。

今回は、YOLOにより物体検出およびカメラ画像内の座標値を検出する「物体検出ノード(Darknet_ROS⁷⁾」、YOLOにより得られた結果からドローンの制御信号を導出する「追従制御ノード(tello_tracking)」、導出された制御指令通りドローン进行操作及びドローンのカメラ映像を伝送する「操作・映像伝送ノード(tello_driver⁸⁾)」の3つのノードを構築することにより、物体追従飛行を可能とした。追従飛行システムのノード間通信の関係を図3に示す。

Darknet_ROSはDarknetにより検出されたクラス名やBoundingBoxの位置情報などをトピックとして送受信できるように変換するもので、GitHubに公開されている。本システムでは、「/darknet_ros/found_object」というトピックに検出したクラス名とBoundingBoxの

位置情報を送信している。

tello_driverノードは、カメラの映像やドローンの操作に必要な信号をROSのトピックとして送受信できるように変換するものであり、こちらもGitHubに公開されている。「/tello/cmd_vel」というトピックにx軸、y軸、z軸の移動速度及びyaw方向の回転速度を入れ込み、tello_driverノードが受信することによってドローンの動作が可能となる。また、「/tello/image_raw」のトピックからドローンのカメラ映像を受け取ることができる。ドローンの座標軸を図4に示す。本来はz軸にカメラを取り付ける必要があるが、今回は、x軸に取り付けられた内蔵カメラを用いた。

物体追従の制御は、自作した「tello_tracking」ノードで行っている。YOLOで得られたBoundingBoxの位置と大きさを一定に保つようにドローンを制御した。今回使用したドローンの映像が960×720[px]なので映像の中心座標(480,360)を目標値として、BoundingBoxの中心が移動するようにドローンのz軸、yaw方向を制御した。また、BoundingBoxの大きさwを一定に保つようにドローンのx軸を制御した。今回はYOLOを使用して追従飛行が可能か否かの確認が目的であるため、精細な制御はせず、比例制御のみ行った。制御に必要な計算結果を「/tello/cmd_vel」トピックに与えることで追従飛行が可能となる。制御のイメージを図5に示す。

2. 3. 2. 物体追従飛行実験

2.3.1.項で示したシステムに対して飛行実験をした。実験の様子を図6に示す。実験にあたって、追従する物体をボトル(bottle)に設定した。学習モデルは既存のyolov2-tinyを使用した。得られたBoundingBoxの位置座標がカメラ映像の中心(480,360)になるように維持しながら、BoundingBoxの大きさw_{bottle}が120になるまで接近または後退するように設定した。実験の結果を図7に示す。YOLOの認識結果から、x軸、z軸及びyaw方向の制御ともに設定値付近を維持しながら飛行できることを目視で確認した。

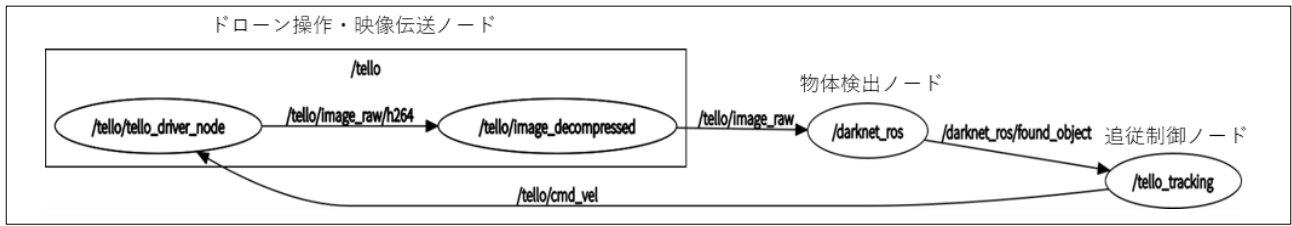


図3 ノードの関係

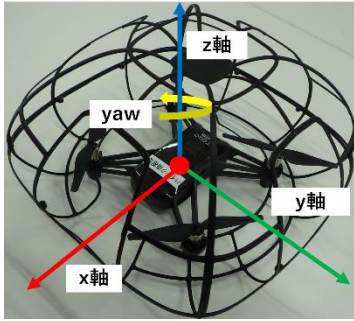


図4 ドローンの座標軸



図6 実験の様子

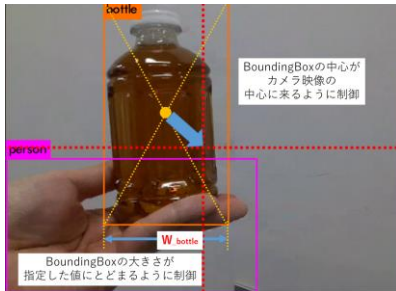


図5 ボトル追従飛行制御イメージ

2. 3. 3. ブレード追従飛行実験

2. 2. 2. で得られた学習モデルを使用してブレードの追従飛行実験をした。

実際に風車のブレードを点検する際は、点検するブレードを真下に向けて固定した状態で点検を行う。真下に位置するブレードの中心を目標として制御した場合、ドローンの位置が少しずれるだけでもブレードの見え方が大きく変わってしまう、支柱と被ってしまいブレードが検出できないといった問題が想定される。そのため、真下に向けたブレード以外の2つのブレードを認識し、その中心位置を算出し制御した。x軸の制御については片方のブレードの BoundingBox の大きさ w_{blade} が 120 になるように設定した。また、カメラ映像にブレードが2つ写っていなかった場合には支柱に一番近いブレードの座標値をもとに制御するよう変更した。制御のイメージを図8に示す。

今回は屋内用ドローンを使用したため、実験室内で風車の画像に向かって追従飛行する形で実験した。結果を図9に示す。本研究で構築した学習モデルにおいても YOLO の認識結果から、z 軸及び yaw 方向いずれも設定値付近を維持し飛行できることを確認した。しかし、x 軸については、制御値から外れる結果となった。

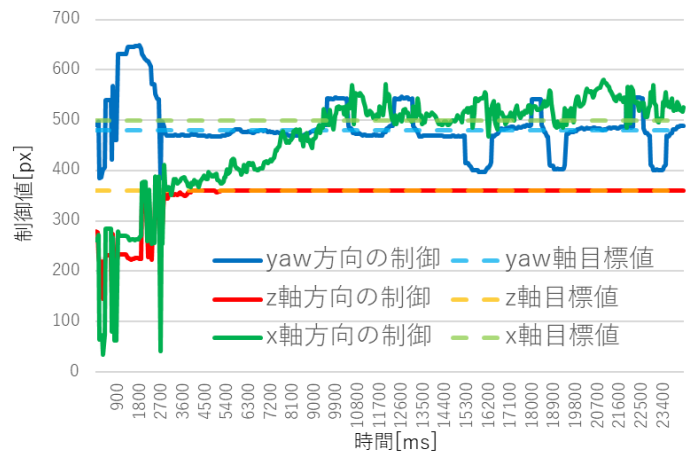


図7 ボトル追従飛行実験結果

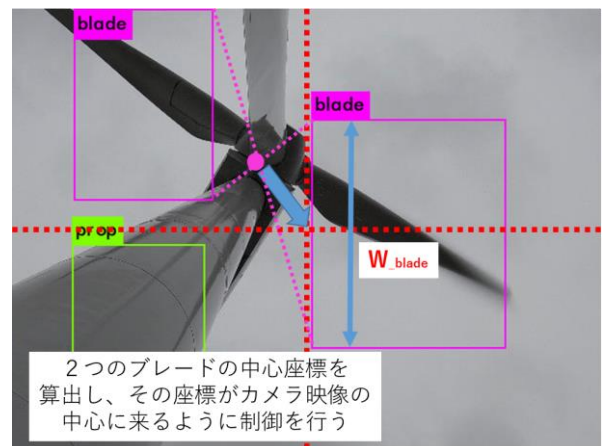


図8 ブレード追従飛行制御イメージ

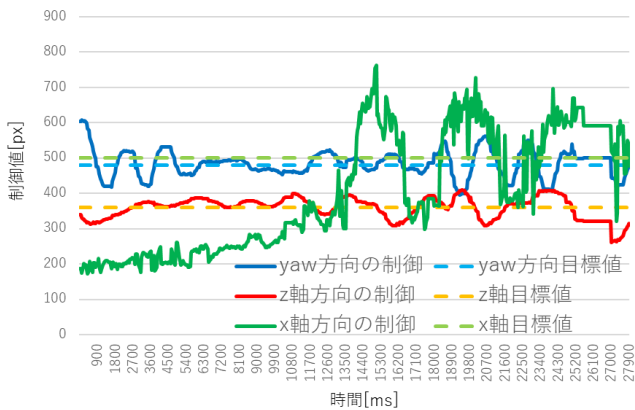


図9 ブレード追従飛行実験結果

3. 考察

3. 1. YOLOによる物体認識

風車のブレードを検出するために、FREAにてデータを収集し、学習させることでブレード及び支柱の検出が可能となった。しかし、今回の学習モデルではFREAの風車のみで学習モデルを作成したため、検出できない風車がでてくるとも想定される。そのため、今後も様々な風車のデータを収集し学習モデルを更新する必要があると考えられる。また、天候による背景の違いや陰によるブレードの色の違いによって検出できない場合も想定される。使用環境を想定して様々な条件で学習する必要があると考えられる。

3. 2. 物体追従飛行

2. 3. 3. 項で x 軸が設定値から外れた原因は BoundingBox の大きさ自体が小刻みに変動し、接近及び後退を繰り返す飛行をしたためだと考えられる。飛行を安定させるために BoundingBox の挙動にロバストな制御アルゴリズムを検討する必要がある。

また、今回は比例制御のみで制御したが、実際は風やドローンの振動などの外乱が発生すると考えられるため、PID 制御などの外乱に強い制御手法についても検討していく必要がある。さらにブレードがカメラの撮影範囲から外れた場合の動作や風車が複数台映り込んだ場合の動作、断線点検センサをブレードに接触させる際の微調整の方法など追従機能周りの動作を固めていく必要がある。

実際の風車で追従飛行を行う場合はブレードの見え方がドローンの移動に伴い、変わっていくため 2. 3. 3. 項で得られなかった問題が生じることも考えられる。そのため、実際に屋外用ドローンで実証試験をする必要があるが、その場合には屋外用の点検ドローンで使用できるように tello_driver ノードに代わる新たなノードを構築する必要がある。

4. 結言

本研究では、ドローン搭載のカメラで風車のブレードを検出し、ブレード付近まで自律飛行する機能を開発した。ブレードの検出は、リアルタイムで物体検出が可能な YOLO を使用し、ブレード検出の学習モデルを構築し評価した。また、YOLO により検出したブレードの座標値をカメラ画像の中心に保つように制御することで常に物体をカメラで捉えるように飛行することが可能となった。さらに、BoundingBox の大きさを一定に保つように制御することで、検出した物体に自動で接近及び一定の距離を保つ飛行が可能となった。

これにより、風車が単体で設置されている場合は YOLO を用いてカメラ画像から物体を検出し、追従飛行することで、風車点検用ドローンが自律飛行できる可能性を示すことができた。今後は、ブレードと断線点検センサを接触させる際の微調整の方法、ブレードがカメラの撮影範囲から外れた場合や風車が複数映り込んだ場合の制御を検討する必要がある。

謝辞

本研究を行うにあたり、学習モデルのデータ収集にご協力を賜りました国立研究開発法人産業技術総合研究所福島再生可能エネルギー研究所様に深謝申し上げます。

参考文献

- 1) Joseph Redmon. "YOLO:Real-Time Object Detection".<https://pjreddie.com/darknet/yolo/>.(参照 2022-1-17)
- 2) J. Redmon, S. Divvala, R. Girshick and A. Farhadi. "You only look once: Unified, real-time object detection." arXiv preprint arXiv:1506.02640,2015.
- 3) DJI Tello edu <https://m.dji.com/jp/product/tello-edu>(参照 2022-1-17)
- 4) "ROS.org | Powering the world's robots". Open Source Robotics Foundation. <http://www.ros.org/>, (参照 2022-1-17).
- 5) GitHub. "pjreddie/darknet". Convolutional Neural Networks. <https://github.com/pjreddie/darknet>. (参照 2022-1-17)
- 6) GitHub. "puzzledqs/BBBox-Label-Tool".<https://github.com/puzzledqs/BBBox-Label-Tool>.(参照 2022-1-17).
- 7) "darknet_ros -ROS Wiki ".http://wiki.ros.org/darknet_ros. (参照 2022-1-17)
- 8) "tello_driver -ROS Wiki ".http://wiki.ros.org/tello_driver(参照 2022-1-17)