

ドローン空撮の再撮影システムに向けた良・不良画像判定システムの高速化

富田 洋文* 中山 恵介* 吉田 英晴** 小松 優陽*** 平間 毅***

1. はじめに

ドローンを用いた空撮は、橋の裏や建物間など目視で確認できない箇所の点検に活用されている。このドローンによる点検では、対象となる建造物周辺を空撮し、得られた画像をもとに建造物のひび割れなどの破損等を確認する。

画像品質は、画像解析技術を用いた一般的な点検の場合と同様に、ドローン空撮における点検の場合でも重要である。収集したデータを確認する際、ブレや極端な明暗などがある不良画像では、破損等の検出ができない場合がある。特に、本研究で対象とする洋上風力発電設備のブレードの点検は、撮影条件が過酷であることから、画像品質確保は大きな課題である。

図1に、本研究が最終目標とする「ドローンが自動で不良画像判定・再撮影するシステム開発」のフローを示す。

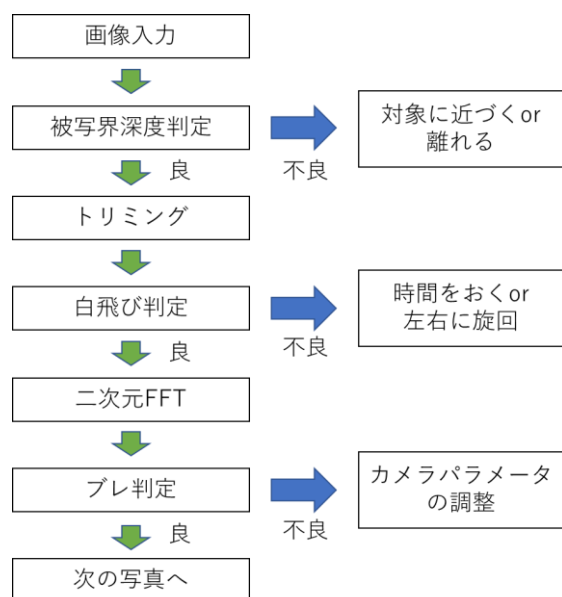


図1 良・不良画像判定システムのフロー

洋上風力発電設備の点検では「風力発電設備の非稼働時間の短縮化」と「洋上の天候の変化」を配慮し、短時間での空撮による点検が求められる。想定するブレードの大きさから、空撮画像1枚の画質判定目標時間は1秒以内である。しかし、構築した開発用PCベースのシステムでは、5280×2970 pixの画像1枚の判定に8秒程度の時間を要するため、判定時間の短縮が課題となった^{1),2)}。

2. 目的

本稿の目標を「ドローンに搭載する小型計算機で、空撮画像1枚あたり1秒以内の判定を行う」と設定した。この高速化を実現するため、PCベースで構築したシステムをドローンに搭載する小型計算機に移植し、処理速度が遅い要因に対処することにした。

3. 研究内容

3.1 ドローン搭載用小型計算機への画質評価システムの移植とその際の実行結果

本研究ではドローン(DJI 社製 Matrice 300 RTK³⁾)を使用し、再撮影システムを構築した。また、ドローンを自律飛行させるため、ドローンに搭載する小型計算機として Jetson Nano Developer Kit⁴⁾を採用した。その採用理由は、Linux 環境をベースとし、軽量かつ低電力で動作するためである。

これまでに構築した画質評価システムは、処理性能が高い Windows PC ベース (ALIENWARE M15 R7) で開発されている。しかし、小型計算機は PC と比較して処理能力が低いので、より長い処理時間を要することが予想される。そのため、小型計算機への移植には開発言語からの見直しを実施した。

PC ベースのシステムはプログラミング言語 Python で構築しているが、プログラミング言語 C++ で構築することで、処理時間の短縮を試みた⁵⁾。

小型計算機かつ C++ で構築したシステムで実行したが、画像1枚あたり約15秒の処理時間が必要となり、さらなる高速化に向けて良・不良画像判定システムのアルゴリズムの改善が必要である。

3.2 画質評価システムの処理速度向上のためのアルゴリズムの改善

図2に良・不良画像判定システムの処理速度の高速化実現のフローを示す。システムの処理速度を改善するために、機械学習モデルを軽量化して、1度の判定時間を短縮した。さらに判定回数の削減とプログラムの不必要箇所の省略等を検討した。これにより良・不良画像判定システムの処理時間を画像1枚あたり平均1.0秒にすることが可能となった。

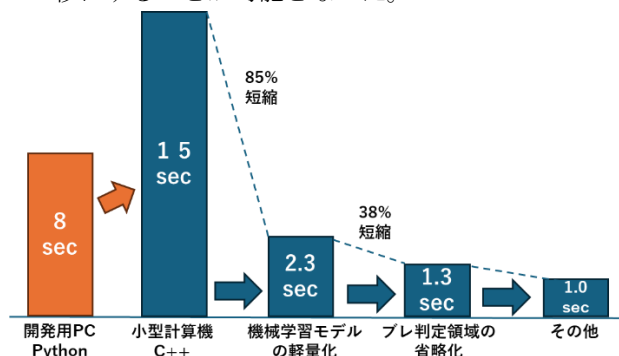


図2 システムの高速化の結果

3.2.1 機械学習モデルの軽量化

これまでの機械学習モデルはニューラルネットワークを使用している。このニューラルネットワークを構成する各要素のパラメータは、精度を優先して設計されており、計算コストまでは考慮されていない。その

ため、小型計算機では機械学習モデルに 320 pix × 320 pix の画像 1 枚を入力した場合、約 0.2 秒の処理時間を要した。

このため、全体の層の削減や全結合層（表 1 の Affine）の結合数の削減といったニューラルネットワークの構成及び各要素のパラメータを調整し、機械学習モデルの精度（ブレ画像の不良判定率が 100%、良画像の良判定率が 99%以上）を維持しながら、計算コストの軽量化を図った。

表 1 に、軽量化したニューラルネットワークの構成を示す。その結果、320 pix × 320 pix の画像 1 枚あたり約 0.03 秒と処理時間が約 85%短縮された。

表 1 軽量化したニューラルネットワークの構成

層	内容	パラメータ	サイズ
-	Input	モノクロ(1)	1,320,320
1	Convolution	KernelShape:(5,5) OutMap:16	16,316,316
	MaxPooling	Shape:(4,4)	16,79,79
	ReLU		16,79,79
2	Convolution	KernelShape:(3,3) OutMap:16	16,77,77
	MaxPooling	Shape:(2,2)	16,38,38
	ReLU		16,38,38
3~4	2層と同様	3層目 OutMap:32 4層目 OutMap:32	32,19,19 32,9,9
5	Affine	OutShape:8	8
	ReLU		
6	Affine	OutShape:5	5
	ReLU		
7	Affine	OutShape:2	2
	Softmax		2
-	CategoricalCross Entropy		1

3.2.2 ブレ判定領域の省略化

画質評価システムでは空撮画像 1 枚を 320 pix × 320 pix でトリミングし、風力発電設備のブレード部が映されている部だけを、白飛び検出とブレ判定している。このブレ判定には、前述した機械学習モデルを使用しており、処理時間のほとんどがこのブレ判定を占めている。そのため、機械学習モデルを軽量化しても、画像 1 枚あたり数秒の処理時間を要してしまう。そこで、ブレ判定に使用するトリミング画像を、一部省略化できないかを検討した。

これまで、評価用に撮影した空撮画像内でブレードが映されている領域すべてについて、ブレ判定を実施していた。しかし、ブレはドローンの揺れ、またはブレード自身の揺れにより発生するものであり、ブレードの一部のみでブレが発生することは無い。このことに着目し、図 3 に示すようにブレードと認識したトリミング画像の端部分のみを判定をすることで、判定回数を削減し、処理時間を短くした。

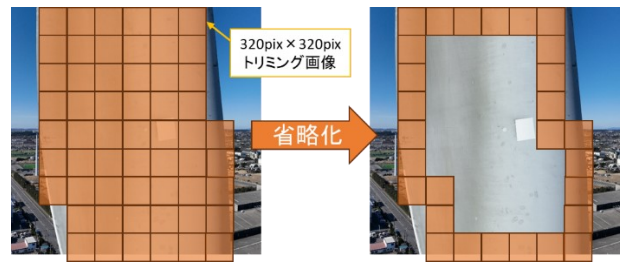


図 3 ブレ判定処理の省略化の例

この削減処理により、空撮画像 1 枚につき平均 45 枚の判定回数を要していたものが、平均 28 枚にまで減少し、処理時間が約 38%短縮された。

さらに、白飛び判定に必要なヒストグラムの計算の不必要な部分の削除や、各判定における過剰な反復処理を削減した。その結果、最終的に、空撮画像 1 枚につき平均 1.0 秒にまで処理時間を短くできた。

4. まとめと今後の課題

本研究では、ドローンによる空撮画像を用いた良・不良画像判定システムにおいて、ドローンに搭載する小型計算機へのシステムの移植と、そのシステムの高速化を実現した。

具体的には、開発言語を Python から C++にし、機械学習モデルの軽量化により約 85%、ブレ判定領域の省略化により約 38%の処理時間を短縮した。

その結果、小型計算機による良・不良画像判定が空撮画像(5280×2970 pix)1 枚につき平均 1.0 秒で処理することを実現した。

今後、本システムとドローンの自律飛行システムを組み合わせ、再撮影システムの開発及び評価実験を実施する予定である。

なお、本研究は令和 4 年度から令和 8 年度まで文部科学省特別電源所在県科学技術振興事業「次世代メンテナンスビジネスに向けたドローンの高機能化に関する試験研究事業」の一環として実施している。

5. 参考文献等

- 1) 富田洋文、中山恵介、吉田英晴、平間毅、ドローン空撮の再撮影システムに向けた良・不良画像判定システムの構築、茨城県産業技術イノベーションセンター研究報告、51、17-20、2024
- 2) 富田洋文、平間毅、中山恵介、吉田英晴、小松優陽、行武栄太郎、画質評価付きドローン空撮システムのモデル構築とその評価、第24回システムインテグレーション部門講演会（SI2023）、3626-3629、2023
- 3) <https://enterprise.dji.com/jp/matrice-300>
- 4) <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano-developer-kit/>
- 5) Farzeen Zehra, Maha Javed, Darakhshan Khan, and Maria Pasha, Comparative analysis of c++ and python in terms of memory and time, Preprints, 2020.