

UAV による空撮画像からの碍子検出 Detection of insulators from UAV images

花田 智生[†]
Tomoki Hanada

太田 寛志[†]
Hiroshi Ohta

クルモフ バレリー[†]
Valeri Kroumov

1. はじめに

送電鉄塔における碍子の点検は高所かつ高電圧下であり作業負担が大きいことからこの負担を軽減することが求められている。近年では、無人航空機 (UAV) の発展に伴い、UAV を用いて自動撮影する研究が盛んにおこなわれている。碍子の点検に自動撮影を利用できれば、作業の負担軽減が期待できる。点検を支障なく行うには、UAV による自動撮影において、点検対象の碍子が画像内に大きく収まるようにカメラの姿勢を調整する必要がある。つまり、画像中の碍子の領域の場所を正確に検出する必要がある。そこで本研究では、DNN (Deep Neural Network) を用いた物体検出アルゴリズムの一つである YOLO[1] を使い、碍子を端部と中間部の 2 クラスで検出する方法を提案する。

2. YOLO による碍子の検出

YOLO を用いて碍子を 1 クラスで検出した場合、得られるクラス確率およびバウンディングボックス (以下、bbox と略す) の情報からは、検出した領域が物体の全体か一部分かを区別することは難しい。そこで提案手法では、碍子を端部と中間部の 2 クラスに分けて認識を行う。これにより、中間部の両側に端部が検出されているか否かによって、碍子の全体か一部分かの区別を行うことができる。碍子の中間部と端部の認識の例を図 1 に示す。緑枠が中間部、紫枠が端部を示している。

3. 使用した学習用画像とデータ拡張

本研究では、YOLO の学習を行うために、碍子が写った原画像 544 枚を用意し、それに対しデータ拡張を行った。データ拡張は物体認識の学習を行う際に、学習に使用する画像に対し行うことで性能が向上することが知られている。本研究ではデータ拡張の手法として、回転 (45°, 90°, 135°, 180°, 225°, 270°, 315°)、反転、平滑化、ガウシアンノイズ、ソルト&ペッパーノイズ、コントラスト増加、コントラスト減少、色温度変化、明度変化、情報欠落を行った。それにより、学習用画像の枚数は 67896 枚となった。データ拡張を導入する前と比べると、AP (Average Precision) は端部が 1.39%、中間部は 1.68% 向上した。この学習用画像を用い YOLO で 30000 回学習した結果、AP は端部が 40.26%、中間部は 49.37% となった。

4. 碍子全体の検出アルゴリズム

YOLO によって得た碍子の端部と中間部の bbox の情報を統合して、ターゲットとなる碍子一つを画像から検出する。提案手法では、最も手前にある碍子が撮影対象であることを想定し、画像内で最も大きく撮像されている碍子を

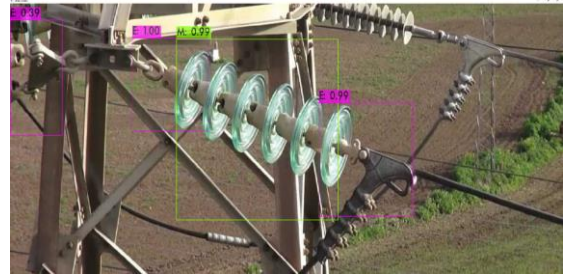


図 1 碍子の認識例

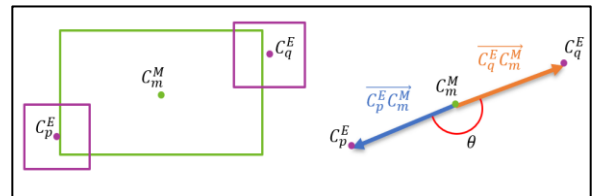


図 2 中間部と端部が直線に並ぶ

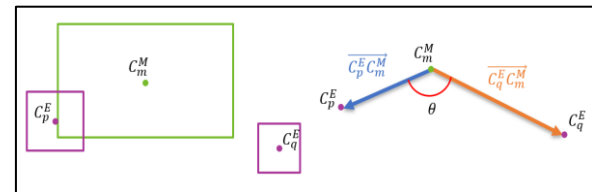


図 3 中間部と端部が直線に並ばない

撮影する。次の手順を繰り返す。

- (1) YOLO を用いて碍子の端部と中間部を検出する。
- (2) 検出した中間部に対し、ターゲットの中間部が未決定の場合は (a) を、決定済み場合は (b) を実行する。
 - (a) bbox の面積が最大のもをターゲットとして決定する。
 - (b) bbox の中心点が画像の中央から最も近いものを新たにターゲットとして決定する。
- (3) すべての端部において、ターゲットの中間部の両側に位置する適切な二つの端部を探索する。
- (4) 適切な端部があれば (a) を、なければ (b) を実行する。
 - (a) ターゲットの中間部と両側の端部を合わせて撮影対象の碍子とし、できるだけ大きく写すよう、カメラ姿勢修正のための情報を出力する。
 - (b) ターゲットの中間部の bbox の中心点が画像の中心になるようにカメラ位置修正のための情報を出力する。

[†]岡山理科大学大学院 Okayama University of Science Graduate School

検出した端部と中間部をそれぞれ E_i ($1 \leq i \leq I$)、 M_j ($1 \leq j \leq J$) と表す。なお、 I と J は画像から検出した端部と中間部の個数である。また端部 E_i について、bbox の幅と高さをそれぞれ w_i^E と h_i^E とし、その中心点を $C_i^E(x_i^E, y_i^E)$ と表す。同様に、中間部 M_j の bbox の幅、高さ、中心点をそれぞれ w_j^M 、 h_j^M 、 $C_j^M(x_j^M, y_j^M)$ と表す。検出アルゴリズムの手順 (2) では、ターゲット候補となる中間部 M_m を定めるために $m = \operatorname{argmax}_{1 \leq j \leq J} (w_j^M \cdot h_j^M)$ を求める。一度目は手順 (2) の (a) を実行するが、手順 (4) で両端にあたる適切な二つの端部が見つからず手順 (4) の (b) を行っていた場合は手順 (2) の (b) を実行する。手順 (3) では、ターゲットとなる中間部 M_m の中心点とその両端に該当する二つの端部の中心点は直線上に並ぶと想定する。ここで、二つの端部 E_p, E_q ($p \neq q$) について、

$$\cos \theta = \frac{C_p^E C_m^M \cdot C_p^E C_m^M}{|C_p^E C_m^M| |C_p^E C_m^M|} \quad (1)$$

を満たす p と q のすべての組み合わせの中で二つのベクトルの長さの和 D が最小となるものを求める。

$$D = |C_p^E C_m^M| + |C_p^E C_m^M| \quad (2)$$

図 2 に中間部と二つの端部の中心点が直線に並んでいる一例を示す。同様に、図 3 に中間部と二つの端部の中心点が直線に並んでいない一例を示す。図 3 のように二つの端部の組み合わせに別の碍子の端部が含まれていると、二つのベクトルがなす角 θ と 180° との差が大きくなり誤った組み合わせであると判断できる。手順 (4) で適切な端部がなかった場合、撮影対象とした碍子の全体が画面に納まっていないと判断できるため、図 4 のようにカメラ位置を修正させるための距離や方向といった情報を計算し、出力する。この情報に基づいて、UAV の姿勢や、カメラを搭載しているジンバルの向き、カメラのズーム倍率などが制御されることを想定している。

5. 性能評価

提案した検出アルゴリズムの性能を評価するために評価用画像を用意し、それに対しアルゴリズムを実行した。各評価画像には碍子一つの全体と周りの背景が写っている。評価用画像の枚数は、碍子を上方から撮影した画像が 86 枚、横方向から撮影した画像 72 枚である。カメラ位置を調整する動作を模擬的に実現するため、評価画像に ROI (Region of Interest) を設定し、提案手法によって得られたカメラ位置の修正情報に基づいて、この ROI を移動させることにする。また、アルゴリズム実行の際の ROI の初期位置を、碍子を中心に碍子の一部が写るように上下左右斜めにずらすことで、画像 1 枚で 9 パターンの評価を行う。実験結果として、表 1 と表 2 にそれぞれ上方からと横方向から撮影した碍子の画像にアルゴリズムを実行した結果を示す。表 1 と表 2 に示したように、提案したアルゴリズムによって画像から碍子をおよそ 85.4% の成功率で検出することができた。図 5 に検出が成功した例と失敗した例を示す。図 5 (b) では、碍子の端部の背景にある鉄塔と色が近かったことから、中間部の鉄塔側の端部が YOLO で検出できなかったため検出に失敗した。検出の失敗の多くは YOLO で碍子の端部や中間部が検出できなかったことが原因であっ

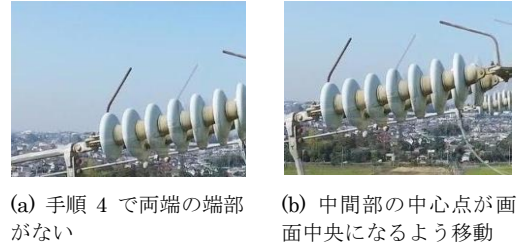


図 4 検出アルゴリズムの手順 4 (b) でカメラ位置を修正した例

表 1 上方から撮影した碍子の検出成功率

画像枚数	検出成功数	検出失敗数	成功率
72	554	94	85.5%

表 2 横方向から撮影した碍子の検出成功率

画像枚数	検出成功数	検出失敗数	成功率
86	660	114	85.3%

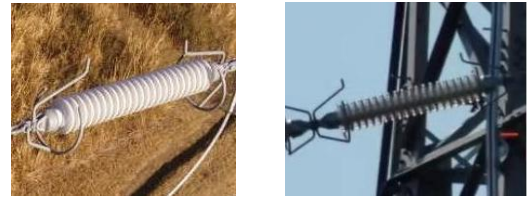


図 5 碍子検出の成功例と失敗例

た。また、想定撮影範囲よりも碍子が大きく画像内に入りきらないことによる失敗もあった。しかし、これは実際にはカメラのズーム倍率を下げるか、UAV を碍子から離すことで解決できると考えられる。

6. まとめ

本研究は画像内に碍子全体が含まれているか否かを判別するために、まず、YOLO を用いて碍子を端部と中間部の 2 クラスに分けて検出を行い、その後、碍子の端部と中間部の情報を統合して碍子の全体を検出する方法を提案した。評価用画像に対して実際にアルゴリズムを実行し、碍子の全体を画像から抽出できることが確かめられた。また性能評価の結果、提案手法は碍子全体を約 85% で検出できることが分かった。今後は碍子の検出失敗が減るように、YOLO の学習方法の改善や YOLO 以外のモデルの有効性について検討を行う。

参考文献

- [1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv:2004.10934 (2020)