

機械学習を用いた空洞内面欠陥自動検出ソフトの開発

DEVELOPMENT OF AUTOMATIC CAVITY DEFECT DETECTION SOFTWARE WITH MACHINE LEARNING

荒木 隼人 ^{*A)}, 浅野 峰行 ^{B)}, 早野 仁司 ^{A)}, 飯竹 真之 ^{C)}

Hayato Araki ^{* A)}, Mineyuki Asano ^{B)}, Hitoshi Hayano ^{A)}, Masayuki Iitake ^{C)}

^{A)} High Energy Accelerator Research Organization (KEK)

^{B)} NAT Co., Ltd.

^{C)} S-Works

Abstract

Defects near the welding bead of SRF cavities sometimes limit the performance of the cavity. As a countermeasure, inner surface inspection camera and local grinding machine are developed to find out and remove the defects on the inner surface of the cavities. We are developing software to automate this process with machine learning object detection methods; Faster R-CNN and YOLO v5. By utilizing the inspection data at KEK over the past 10 years as a teacher data, our software obtained enough performance compared to human inspection. We will report the structure of this software, the status of development, and future plan to improve its performance.

1. はじめに

超伝導空洞の内面に、常伝導の不純物や溶接欠陥のような大きな RF 損失を生じる領域が存在すると、ジュール熱によって周辺部が超伝導転移温度を超え、クエンチしてしまう。また、不純物がなくても幾何学的な凹凸が存在すると表面磁場が局所的に増大し、臨界磁場を越えることでクエンチの原因となる。このような領域のことを欠陥 (defect) と呼び [1], 超伝導空洞の性能を制限する大きな要因になる。特に、溶接部付近の空洞内面に存在する欠陥はフィールドエミッションやクエンチにより最大加速勾配を制限するが、これを克服するために開発されたのが空洞内面検査装置である [2]。この装置は空洞内面を撮影して原因となる欠陥を発見し、局所研磨装置で除去することで性能制限を克服することができる。実際に内面検査装置で撮影された欠陥の写真を Fig. 1 に示す。

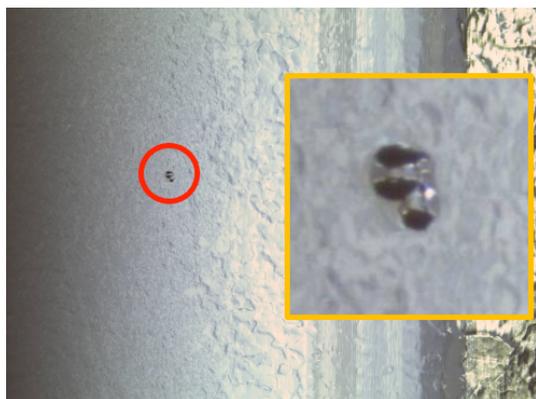


Figure 1: A defect observed with inspection camera.

ILC 用の 1.3 GHz 9-cell 空洞 1 台を検査する場合、撮

影する画像の枚数は約 3,000 枚になる。現状ではこの画像を全て人間が検査することで欠陥を発見しているが、撮影と検査を合わせた処理能力は週に最大 2 台が限度となっていて、ILC ではこの空洞を約 9,000 台量産する必要があり [3], このような規模では全数検査することができない。また、空洞内面の様子は空洞の材料や製造工程、表面処理工程によって変化するため、その中から性能に影響する欠陥であるかを判定するのは人間にとって容易ではない。

今回、撮影された画像から欠陥を自動的に検出するソフトウェアを開発したので、報告する。

2. ソフトウェア

2.1 物体検出アルゴリズム

内面検査装置の欠陥を自動検出する試みは過去にも行われてきた [4, 5]。これらの先行研究は従来手法の物体検出技術を使っているが、近年の機械学習の発展によって新たなアルゴリズムが登場している。今回作成したソフトウェアでは、Faster R-CNN と YOLO v5 という 2 つのアルゴリズムを試した。

Faster R-CNN [6] は、リアルタイム検出として実用性のある検出速度を達成した初めてのアルゴリズムとして 2015 年に発表された。これをベースとして複数のアルゴリズムが更なる高速化と高性能化を目指して発展しているが、これ自身も十分実用的な性能を有しており、また知名度も高いことから多くの実装例がインターネット上で公開されている。

YOLO [7] は Faster R-CNN と同時期の 2015 年に発表されたアルゴリズムで、環境によるが 150 fps を超える高速処理が特徴である。今回は、後継として開発されている YOLO v5 [8] を使用した。

2.2 開発

画像処理や機械学習で使われるライブラリは python で開発されていることが多いため、ソフトウェアの開発言語には python を使用した。また、前節で述べた 2 種類のアルゴリズムを実装するフレームワークとして

* arakih@post.kek.jp

PyTorch [9] を使用した。

試作段階では Google Colaboratory [10] を使っていたが、最終的に誰でも使えるソフトウェアにすることを目的としていたことと、十分な性能を持つ計算機が入手できたため GUI を実装した。実装には PySimpleGUI [11] を使用しており、これによってマウスドラッグによる領域選択等の直感的な操作を少ないコード量で記述し、かつクロスプラットフォーム (Windows, MacOS, LINUX) 対応のソフトウェアとすることができた。このソフトウェアの画面の例として、検出結果の確認と編集を行う画面を Fig. 2 に示す。

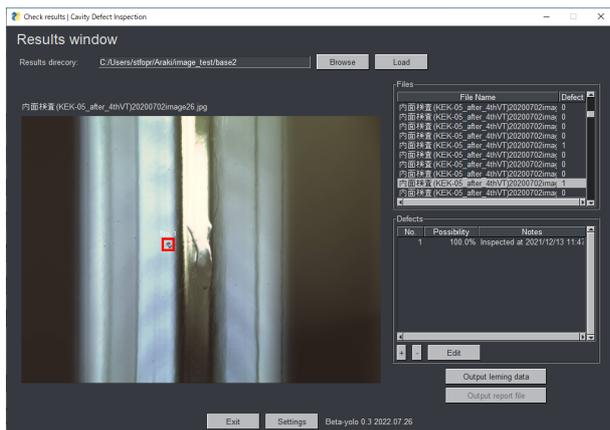


Figure 2: A screenshot of detection result window.

3. 性能評価

3.1 データセットの作成と試験

性能評価用のデータセットとして、KEK での過去 200 回分の内面検査レポートに含まれる約 4,000 枚の画像を使用した。これはプレゼンテーションファイルから抽出したもので、欠陥を含まない画像も多く混ざっている。最も古いものは 12 年前に撮影されたもので、そこから現在に至るまでのカメラの交換による画素数の増加や、空洞製造技術や表面処理技術の向上による欠陥形状の変化を全て含んだデータである。この中からランダムに選んだ 100 枚をテスト用、残りを学習用とした。

まず、テスト用と学習用ともに全ての画像を人間の目で検査し、欠陥と判断した場所を手動でマーキングした。このとき、欠陥が含まれていた画像はテスト用画像のうち 39 枚 (欠陥 75 箇所)、学習用画像のうち約 1,600 枚であった。また、判定の基準が一定になるようこれらの作業は全て同一人物が行った。

次に、作成した学習用データセットを Faster R-CNN および YOLO v5 に学習させた。後者には Albumentations [12] を使用したデータ水増し (data augmentation) が実装されており、前者にも同等のものを適用した。学習は CUDA [13] が利用できる GPU (NVIDIA Quadro RTX 4000) が搭載された計算機で行い、学習時間は両者ともに 100-500 epochs で数時間から数日程度かかった。

以上によって学習したモデルをテスト用画像に適用して検出を行い、その結果から precision と recall を計算し

た。Precision は誤検出の少なさ、recall は見落としの少なさを表す。100 枚の画像を検査するのにかかった時間は両アルゴリズムとも 1 分程度であった。また比較のため、データ作成者とは異なる人間による検査も行った。1 人目の検査者 (Human A) は内面検査の経験が豊富な熟練者であり、2 人目 (Human B) はこれまでに検査の経験がなく簡単な説明を受けただけの初心者である。その結果を Table 1 に示す。

Table 1: Defect Detection Result of Test Dataset

	Precision (%)	Recall (%)
Human A	89.5	44.0
Human B	56.3	46.3
Faster R-CNN	52.7	81.3
YOLO v5	53.1	77.3

この試行では、両アルゴリズムともに precision を 50-60% になるようにパラメータを調整して recall をすることを目的としている。これは実用的な目標としてソフトウェアをスクリーニングとして使用し、最終的な判断は人間に委ねるという使い方を想定しているためである。検査結果の recall にはそれが反映されており、両アルゴリズムともに人間より高い結果となった。また、この precision の目標値は結果的に初心者である B と同程度であり、その上で B よりも明らかこのことから、少なくとも初心者による検査よりは優位性があると言える。

3.2 新造空洞での試験

KEK では 2021 年度末に新しい 9-cell 空洞 KEK-7 の製造を完了した。前節のデータセットはそれ以前の画像から作成しているため、この空洞の情報は含まれていない。よって、このデータセットで学習したモデルにとっては完全に未知のデータとなる KEK-7 の内面検査画像に対して欠陥検出を行った。

KEK-7 は 2022 年 7 月までに 3 回の内面検査を行っているが、このうち従来方法では EP1 後の検査のみ 1 箇所欠陥が発見された。この検査では合計 2,694 枚の画像を撮影し、そのうち欠陥を含むのは 1 箇所をそれぞれ違う角度で撮影した 3 枚の画像だけである。またこの空洞特有の事情として、材料の結晶粒界が目立っている、製造工程でアイリス部の複数箇所欠陥に傷がある、直前の電解研磨の研磨液による気泡の跡があるといったことが挙げられる。

この EP1 後の画像をソフトウェアで全て検査したところ、両方のアルゴリズムが 3 枚の画像に含まれる欠陥を全て正しく検出できた。検出の例として、Faster R-CNN での欠陥検出結果を Fig. 3 に示す。またソフトウェアが欠陥と判断した画像は Faster R-CNN で 246 枚、YOLO v5 で 618 枚であり、元の画像総数の 4 分の 1 以下である。検出すべき欠陥を含む画像は両者とも全て検出できているので、この検出画像数の違いは precision が約 2.5 倍異なることを表している。検出された画像にはアイリス部の傷や研磨液による気泡の跡が含まれていた他、人間の検査者が判断に迷う画像も多く含まれていた。このような結果に対する定量的な評価はまだできていない

が、スクリーニングの目的ではおおむね実用性のある性能を有していると考えている。



Figure 3: Defect detection result of KEK-7 with Faster R-CNN. The marked position was also detected by YOLO v5 and human inspection.

4. まとめと展望

超伝導空洞の内面検査画像から Faster R-CNN と YOLO v5 の 2 種類のアルゴリズムを用いて自動で欠陥を検出するソフトウェアを開発し、性能試験の結果からスクリーニング(人間の検査の補助)としては実用性のある性能を有していることがわかった。

更なる性能向上の手段としては、まず質の良い学習用データセットを作成することを考えている。今回のデータセット作成に使用した画像はレポートからの抽出であることから、明らかな欠陥が含まれるか全く問題ないかという極端な画像が集まっており、スクリーニングで検出したい「欠陥かどうか曖昧な物体」が少ないという問題がある。これに対しては、より多くの画像を偏りなく使用したデータセットを作成することで解決できる。また、基本的な手法であるハイパーパラメータの最適化や転移学習による学習時間の短縮といった手法も今後試したいと思っている。

実用化に向けては、検査レポートの自動作成等の必要な機能を追加しつつ、空洞 R&D の現場で実用上の問題点の洗い出しを行う予定である。

謝辞

試験結果についての助言をいただいた京都大学特任准教授 岩下芳久氏ならびに KEK iCASA SRF グループの皆様、性能比較に協力いただいた KEK 小笠原舜斗氏に感謝いたします。

参考文献

- [1] H. Padamsee, “RF Superconductivity: Science, Technology, and Applications”, Wiley-VCH, 2009.
- [2] Yoshihisa Iwashita, Yujiro Tajima and Hitoshi Hayano, “Development of high resolution camera for observations of superconducting cavities”, Physical Review Special Topics -

- Accelerators and Beams **11**, 093501, 2008; doi:10.1103/PhysRevSTAB.11.093501
- [3] “Recommendations on ILC Project Implementation”, High Energy Accelerator Research Organization (KEK), 2019; https://www.kek.jp/ja/newsroom/attic/20191001_%20ILC%20Project.pdf
- [4] Y. Kikuchi *et al.*, “Automated image capture and defects detection by Cavity inspection camera”, Proceedings of Particle Accelerator Society Meeting 2009, JAEA, Tokai, Nakagun, Ibaraki, Japan, 2009; https://www.pasj.jp/web_publish/pasj6/papers/fpaca09.pdf
- [5] M. Wenskat, “First attempts in automated defect recognition in superconducting radio-frequency cavities”, Journal of Instrumentation **14** P06021, 2019; doi:10.1088/1748-0221/14/06/p06021
- [6] S. Ren *et al.*, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, Advances in Neural Information Processing Systems **28** (NIPS 2015), 2015, pp. 91-99; doi:10.5555/2969239.2969250
- [7] J. Redmon *et al.*, “You Only Look Once: Unified, Real-Time Object Detection”, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788; doi:10.1109/CVPR.2016.91
- [8] https://pytorch.org/hub/ultralytics_yolov5/
- [9] <https://pytorch.org/>
- [10] <https://research.google.com/colaboratory/>
- [11] <https://pypi.org/project/PySimpleGUI/>
- [12] Alexander Buslaev *et al.*, “Albumentations: Fast and Flexible Image Augmentations”, Information **11**: 125, 2020; doi:10.3390/info11020125
- [13] <https://developer.nvidia.com/cuda-toolkit>