

機械学習によるカメラ映像のリアルタイム異常検出

○前野 博也^{A)}、水野 嘉識^{A)}、大場 恒揮^{A)}

^{A)} 核融合科学研究所 技術部

1. はじめに

核融合科学研究所の大型ヘリカル装置（LHD）では、装置本体の周りに 16 個のカメラを設置してプラズマ実験中に監視を行っている。そのうちの 1 台は、加熱装置の 1 つである電子サイクロトロン共鳴加熱装置（ECH）の動作を監視するものである。この装置の動作によっては LHD 内部の壁面に損傷を与える可能性があるが、その際にはカメラの映像中に特徴的な発光が現れる。実験中は ECH の担当者が映像を監視するが、全ての映像を常時監視するのは現実的ではない。そこで、その補助として画像認識 AI を用いて異常検出を行い、担当者に警告を行う仕組みを構築した。

2. 使用機材および環境

映像信号は解像度 640x480、30fps の HDMI 信号である。キャプチャデバイスは一般的な HDMI-USB2.0 変換であるサンワサプライの USB-CVH DUVC2 を用いる。プログラムを動かす PC は Windows 11 で、GPU として GeForce RTX 4070 Super を備える。機械学習に標準的に使用される Python 環境として、Anaconda を利用することとした。

3. 学習モデルの作成

異常検出に機械学習を利用するため、リアルタイム性に優れた物体検出アルゴリズムである YOLO^[1]（Ultralytics 社提供）を利用することにした。

同社からは、人や車といった一般的な物体を検出できる学習済みモデルが配布されている。ただし、今回の検出対象である発光は検出できない。また、発光にも様々なパターンがある。このため転移学習と呼ばれる手法によって、本目的のために特化した専用モデルを作成する必要がある。以下にその手順を述べる。

初めに、学習させるデータを準備する。今回は、過去実験での同一カメラの映像から、異常と判定される画像を学習データとした。プロトタイプであるため、データ数は約 100 枚とした。これらの画像のうちの 1 枚を図 1 に示す。

次に、それらの画像に対しアノテーション（位置



図 1 検出対象となる画像（四角は検出部分）

指定)を行う。画像内の検出対象を矩形で囲み、その座標情報を YOLO フォーマットのテキストファイルとして保存した。

こうして準備したデータセットを、学習用と検証用に 7:3 の割合で分割した。Ultralytics 社が配布している事前学習モデル（yolov8n.pt）を使用し、yolo detect train コマンドによって転移学習を行った。

4. 処理の流れ

モデルの準備後、プログラムを作成した。実験は 3 分または 3 分 30 秒のシーケンスが繰り返されるが、そのうち最初の 2 分 30 秒は準備時間であり、2 分 30 秒後の放電開始タイミングを T=0 と呼ぶ。T=0 からの 10 秒または 40 秒が放電時間であり、それに合わせて検出を行う。検出開始・終了のトリガーは TCP/IP で取得する。

トリガーを受信すると、映像に対してリアルタイム推論を行う。ここで、30fps において 1~2 フレームで消える発光は検出対象に当てはまらないことがわかっているため、同様の検知が 5 回連続した場合のみを検出とした。このために連続するフレーム間で IoU（Intersection over Union）という指標を計算し、指定した閾値を超え続けることを条件とした。

検出と判断したらその画像を保存し、1 秒待機して推論に戻る。それを繰り返し、終了のトリガーを受け取るかタイムアウトしたら、検出した画像を全て表示して待機状態に戻り、次の開始トリガーを待つ。

5. ユーザーインターフェース

検出に対しては、信頼度と呼ばれる AI の判定の確信度を表す指標が付与される。これは 0 から 1 までの値を取り、大きいほど AI が判定に自信を持っていることを示す。プログラム中ではこれに対して閾値（信頼度閾値）を設定し、それを超えていれば検出とするが、その値は可変とするのが一般的である。今回の検出においても、この閾値がどの程度であれば実際に検出したい発光が過不足なく判定されるか不明瞭であったため、プログラム実行時に UI を表示し、閾値を簡易に変更して値を即座に反映できるようにした。この UI のスクリーンショットを図 2 に示す。

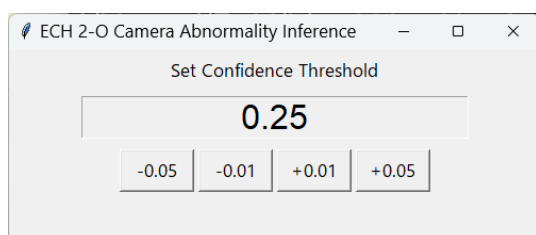


図 2 ユーザーインターフェース

6. 運用結果

LHD の第 26 サイクル実験において、実際に運用を行った。保存された画像を確認すると、想定通りの検出がされていることが確認できた。ただし、誤検出も多くあった。

ECH 担当者からは「光った場所とタイミングが表示されるのは良かった」という声があった一方、「プラズマの可視光で反応している場合が多かった」という意見もあった。

検出結果の一例を図 3 に示す。上の画像は複数箇所を正しく検出している例、下の画像は誤検出の例である。検出位置を示す四角形（バウンディングボックス）に添えられた数字は信頼度、右下の数字は実験のシーケンス番号を示す。下の画像の誤検出がプラズマの可視光に反応しているものである。今回は対処出来なかったが、画像全体の輝度が高いものを除外するようにすれば、このような画像を検出しないようにすることは可能だと考えられる。

7. まとめ

画像認識 AI を用いて監視カメラ映像の異常な発光を検出するプログラムを開発した。検出精度などに改善の余地はあるものの、概ね想定通りの検出結果が得られ、監視補助としての役割を果たすことができた。今後、同様の事例があれば今回の成果を応用することが十分可能であると考えられる。

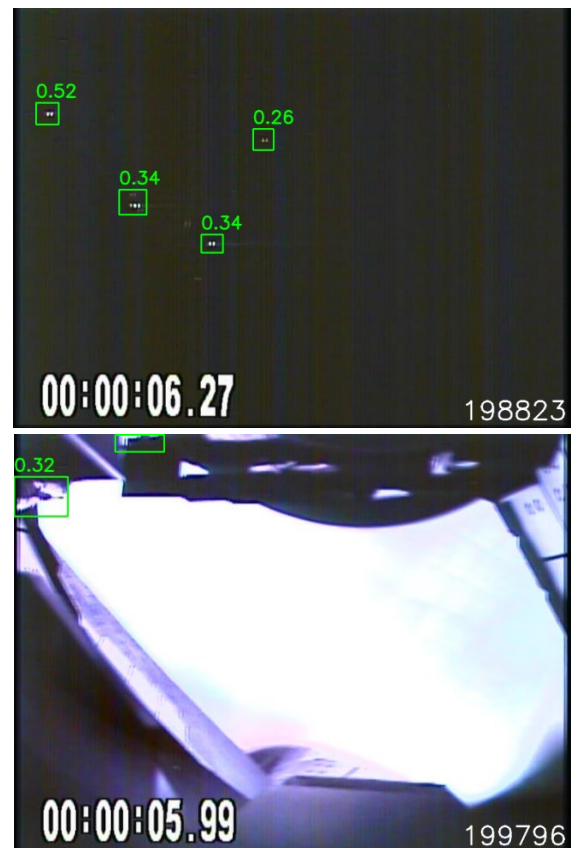


図 3 検出結果

参考文献

[1] <https://docs.ultralytics.com/ja/>