

AI・IoT を活用した県内産業のスマート化推進に関する試験・研究開発

(①ワイヤロープ異常検知システムの開発)

Study on smarter industries of Aomori using AI and IoT
(1. Development of wire rope anomaly detection system)

奥田 雄人

少子高齢化に伴う人口減少に対応するため、これまでも製造業等における AI・IoT 等のデジタル技術の活用による生産性の向上を目指してきたが、コロナ禍によりその重要度が増大した。また、昨今、あらゆる産業において、データ駆動型経営にシフトしながら企業風土やサービス等を変革することで新たな価値を創造し競争上の優位性を獲得していく DX(デジタルトランスフォーメーション)の推進が求められている。

本研究では、クレーン車のワイヤロープ異常（乱巻き）による重大事故を未然に防ぐため、クレーン車搭載カメラから画像データを収集し学習させることで、乱巻きを検知し、運転者に通知することができるシステムを開発した。また、ワイヤロープ異常検知システム[1]として特許を出願し、協力企業へ技術移転を行った。



図 ウインチ部の映像と異常度のリアルタイム出力
(実稼働時は警告音を出力)

1. はじめに

本研究は、令和2年度に県内建設事業者からの相談をきっかけに着手したものである。現状では運転席のモニタからワイヤロープの状態を確認できるが、ウインチ部でワイヤロープの位置が乱れる「乱巻き」の発生に気づかず、乱れたまま巻取りを続けるとワイヤロープがつぶれてしまうことがあり、その度に数十万円を費やしてワイヤロープを全交換する必要がある。

AI 技術による異常検知は製造現場等で用いられているが、本課題のシステム開発においては、時間帯や天候などの影響を受ける屋外という多様な照明条件下で、「乱巻き」という多様なパターンの異常を瞬時に検知することが求められる。

本研究では、既設のカメラを使い、撮影環境が一定でなく、正常・異常ともにパターンが多く、異常データの取得自体も困難という条件下で AI を適用する手法の開発に取り組んだ。また、前述の取り組みを基に、建設事業者等の既存クレーン車に機器を追加することで機能し、クレーン車のワイヤケーブルの破損要因であるウインチ部の乱巻き発生を検知し、ワイヤロープ破損前にオペレータに通知できるシステムを目指した。



図1 クレーン車のウインチ部

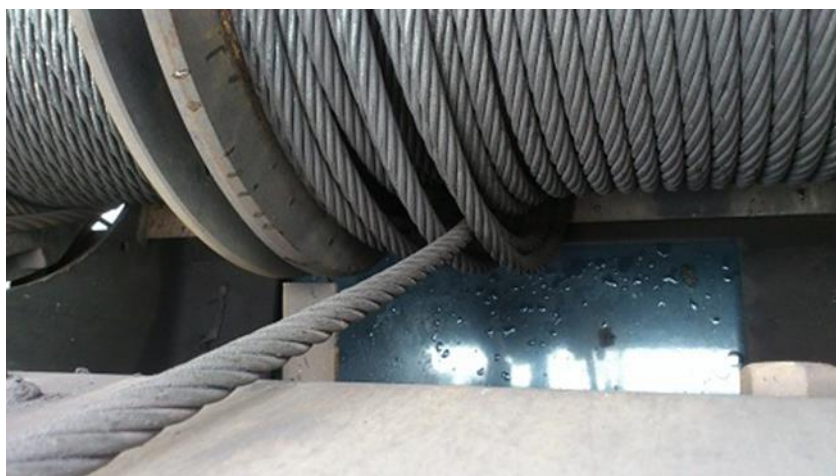


図2 乱巻き発生時（クレーンオペさんのブログ[2]より引用）

2. 実験方法

2. 1. 画像取得方法の確立

ウインチ部の異常をリアルタイムで通知するシステムを構築するためには、判断に使われる信号を常時取得する必要がある。センサ等を追加することで、音や振動といった信号を利用することも考えられるが、電源供給の方法や取り付け位置等を考慮しなければならない。本研究では、より容易に後付け可能なシステムとするため、クレーンに搭載されているカメラを使用することとした。

比較的大型のクレーン車には、ウインチ部やブーム先端にカメラが搭載されていることが多く、運転席のモニタから状況を確認することができる。また、オプションとして、映像を記録するためのレコーダーと、レコーダーをカメラと接続するためのケーブルが販売されている。このケーブルの配線を図3のように変更することで、モニタへの出力を分配し、カメラ画像の取得が行えるかを試みた。

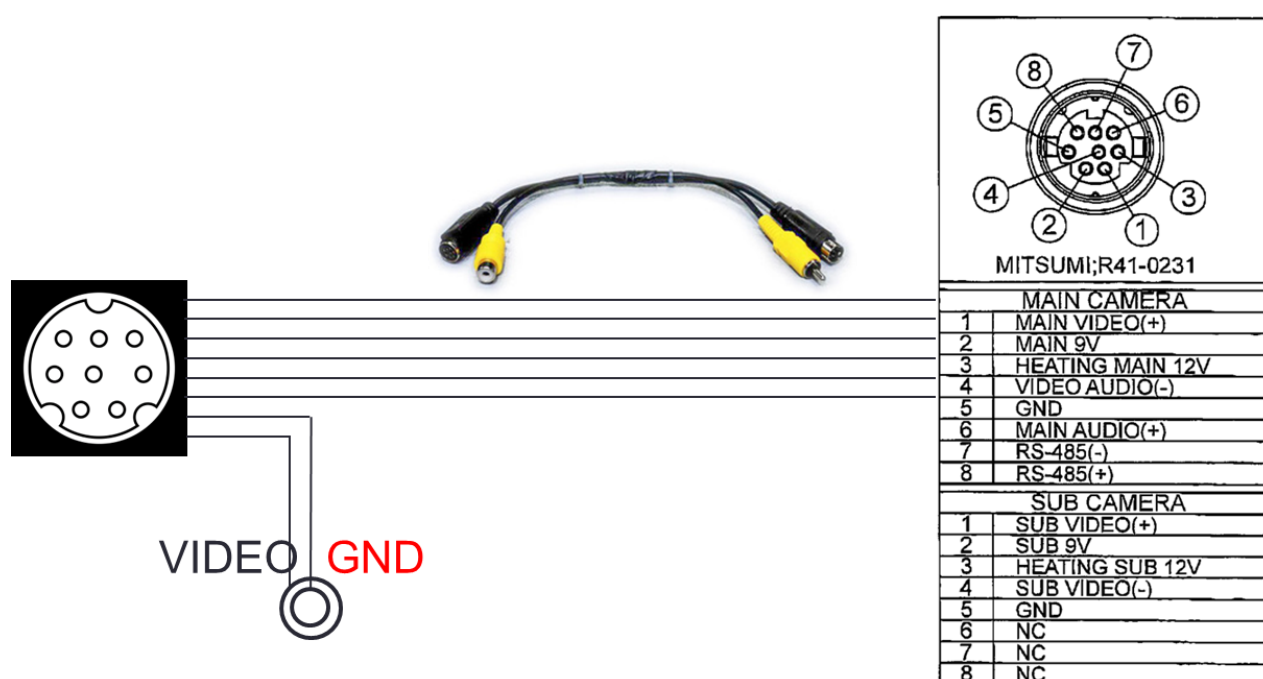


図3 映像取出ケーブルの配線図（クラリオン社から入手）と変更箇所（赤字）

画像の取得には市販のアナログビデオキャプチャを使用した。また、映像を画像に変換するためにFFmpegを使用した。これにより、クレーン車のカメラから画像を取得可能であることを確認した。

2. 2. 異常検知手法の試行

異常検知手法として、AI を用いた手法のほか、画像処理による手法も合わせて検討した。

2. 2. 1. ウェーブレット変換を用いた試行

ウインチ部でワイヤロープの巻き方の乱れを検出したいため、波形の乱れを検出可能なウェーブレット変換を画像に適用し試行した。図4では、横軸1200付近で、周期のピークが縦軸20付近から大きく乖離している様子が見られる。このように画像中の周期の乱れを乱巻きとみなすことで検出できる。

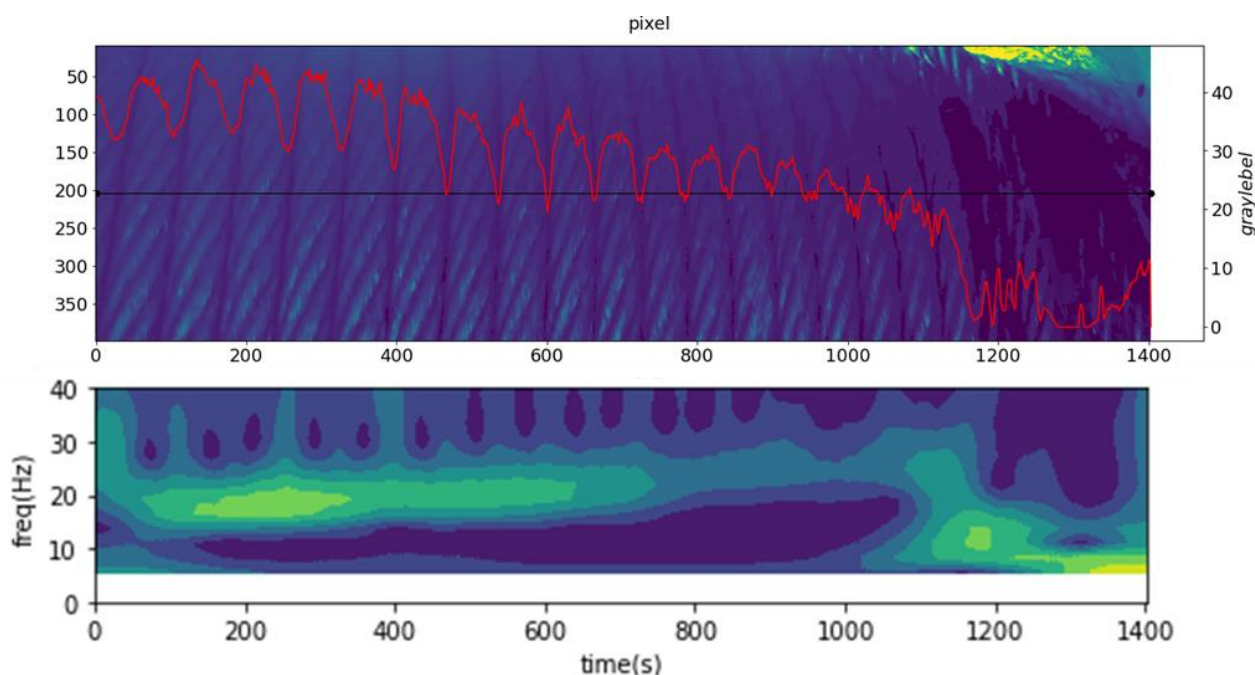


図4 上：輝度値と元画像のオーバーレイ、下：輝度値のウェーブレット変換結果

しかし、前述の方法で乱巻きを検出するためには、カメラの向きやレンズの特性による歪み、ウインチの形状等を考慮した補正を行う必要が生じる。また、屋外環境のため、クレーンの影などの照明条件の変化により、輝度値の周期変化を正しく取得することが難しい。このように、単純な画像処理では判別が難しいため、あらためてAIを用いた手法を検討することとした。

2. 2. 2. Teachable Machine による試行

AI による手法で乱巻きを判別可能かどうか試行するにあたり、Google が提供している Web サービスである Teachable Machine[3]を使用した。学習データには、図 5 のように、ワイヤロープに見立てたケーブルを、綺麗に巻かれた状態 (Class 1) と乱雑に巻かれた状態 (Class 2) の 2 種類を、それぞれ 200 サンプル以上取得し用いた。トレーニングを行い作成された学習済みモデルを使用し、ケーブルを整列させたり乱雑に巻いたりしてプレビュー表示上で確認したところ、概ね良好な判定結果を示したことから、AI による手法を検討することとした。



図 5 Teachable Machine による学習と結果出力

2. 2. 3. オートエンコーダを用いた試行

AI による手法では、Teachable Machine で試行したような教師あり学習手法を用いることが一般的であるが、図 6 に示すオートエンコーダによる手法を用いることで、学習の際に異常データが不要となる可能性がある。そこで、得られた 5000 枚の正常データのみから、乱巻きの検知が行えるかどうか試行した。

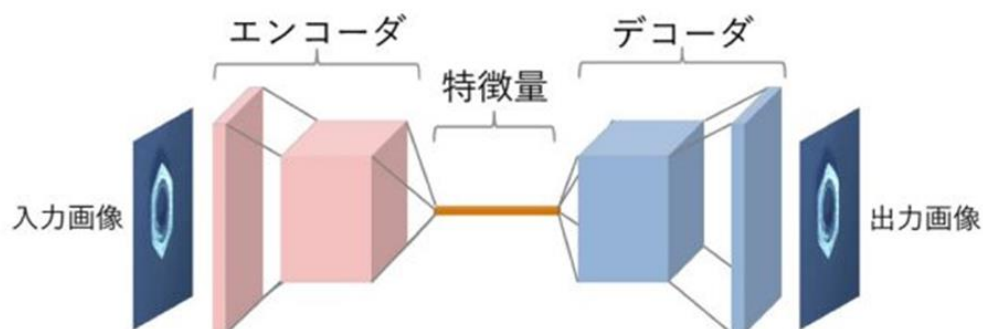


図 6 オートエンコーダ概略 (MathWorks[4]より引用)

結果は、図 7 に示すとおり、乱巻き発生時の異常度が正常時より高くはならず、乱巻きの検出は行えなかった。オートエンコーダにより生成された出力画像を確認すると、入力画像から影が除かれていたことから、乱巻き部分より広範囲に生じる影の影響を強く受けていると考えられる。



図 7 オートエンコーダによる正常／異常画像の異常度の比較

2. 2. 4. 画像処理による特徴強調の試行

前述のオートエンコーダを用いた試行の際、影の影響を減らす工夫として、画像処理による特徴強調が行えないかを試行した。図8は、画像編集ソフト GIMP を用いてグレースケール化と輪郭抽出を行った結果である。ワイヤロープが整列している部分は輝度が下がり、赤枠の乱巻き部分は輝度が上がっている様子が見られる。

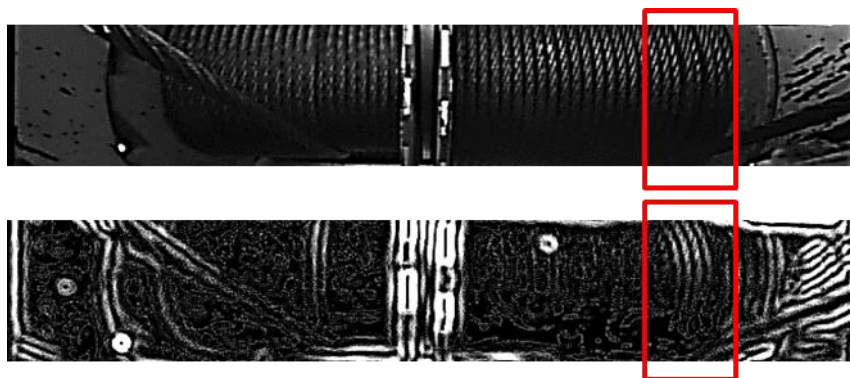


図8 輪郭抽出等を用いた特徴抽出結果

しかし、輝度の変化が生じる箇所には輪郭も抽出されてしまうため、乱巻き部分以外を除くために追加の処理が必要となる。また、ワイヤロープの巻き方向に沿って、ワイヤロープと同じ線幅の影が生じた場合、乱巻きと区別するのは極めて困難と考えられる。そのため、オートエンコーダを用いた手法は断念し、教師あり学習による手法を検討することとした。

2. 3. ニューラルネットワークの構築と学習

前節で説明した画像処理によるアプローチでは、画像の特徴量を人為的に抽出する手法を用いた。しかし近年は、画像の特徴量を大量のデータから自動的に抽出する手法（機械学習）が用いられることが多い。また、ハードウェアの進歩により、多数の層から構成されるニューラルネットワークによる深層学習（ディープラーニング）も現実的に適用可能となりつつある。現在は、深層学習により作成された学習モデルや、学習モデルを搭載したシステムが AI（人工知能）と呼ばれている。

本研究においても、深層学習による手法を用いる。また、ハードウェアの制約によりあらかじめ特徴量の削減を行なったため、表 1 に使用したハードウェア構成、図 9 と図 10 に特徴量の削減方法を示す。なお、学習ツールは、Sony の Neural Network Console[5]を使用した。

表 1 学習モデル作成に使用したハードウェア構成

CPU	Intel Xeon E3-1270v5 (3.60GHz)
Mem	16GB (2x8GB, 2133MHz DDR4)
GPU	Nvidia Quadro P1000 (4GB GDDR5)
SSD	512GB (PCIe 3.0x4, MLC V-NAND)
HDD	1TB Serial ATA 6Gb/s (7200rpm, NCQ)
OS	Windows 10 Pro for Workstations

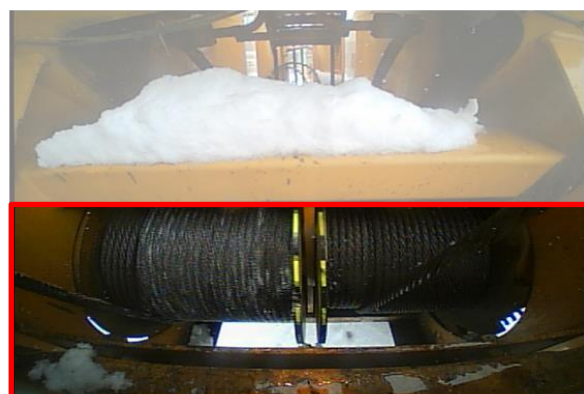
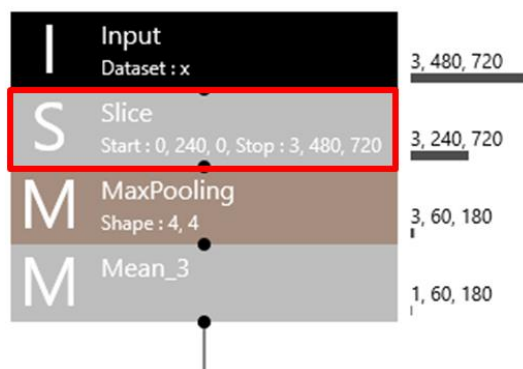


図 9 領域切り取りによる特徴量の削減

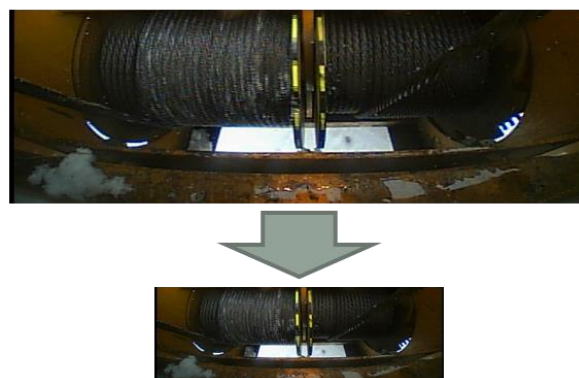
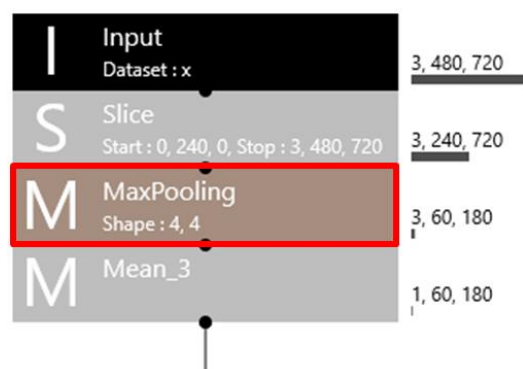


図 10 領域縮小による特徴量の削減

アルゴリズムは、画像データを扱う場合に一般的な CNN(Convolutional Neural Network)を使用し、図 1 2 左のようにニューラルネットワークを構築した。学習データの収集は、協力企業のクレーン車が空く 2 月に一日貸切で行った。また、異常データ収集のため、以下の手順で乱巻きを意図的に発生させた状態のウインチ部を撮影した。

1. フックを地面に下ろす（フックが倒れて壊れないよう慎重に）
→ワイヤが緩む
2. ブームを下げる
→乱巻きがはじまる
3. ワイヤを繰り出す・シーブを動かす
→よりひどい状態になる（破損しないよう最後は元に戻す）
4. ブームを上げる
→乱巻きが直る
5. フックを地面から上げる
→ワイヤが張る

これにより、乱巻き画像約 5000 枚を含む約 13000 枚のウインチ部画像（解像度 720x480 の RGB 画像）が得られた。但し、CNN に入力する画像は、前述の特徴量の削減を行った画像（ウインチが写っている下半分をトリミングし 25%に縮小した画像）である。

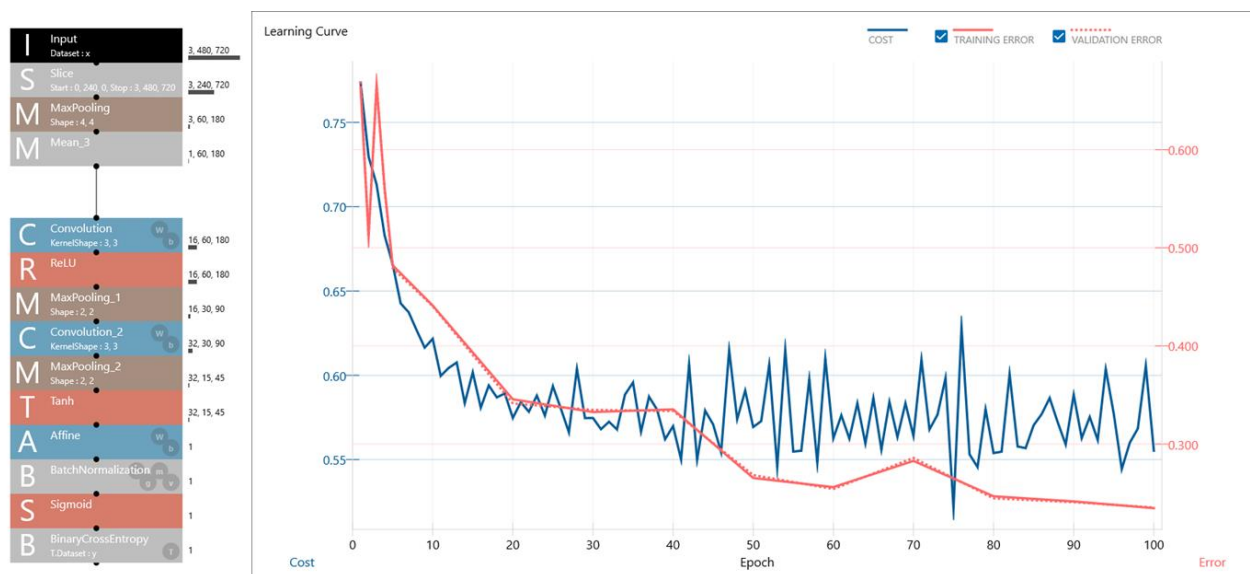


図 1 1 構築したニューラルネットワークと学習曲線

学習曲線は図 1 1 右のとおりとなった。グラフの横軸は Epoch であり、学習の試行回数を示す。

青色のグラフの縦軸は、ニューラルネットワークの出力と実際の出力の差を示す。また、赤実線は訓練データとの誤差（訓練誤差）であり、赤点線は評価データとの誤差（汎化誤差）である。

今回試行した学習は 100Epoch で約 6 時間かかった。また、学習データが少ないため、訓練データと評価データは同じデータセットを使用した。結果、20Epoch 付近までは順調に学習が進んだが、それ以降収束せず、あまり良い学習曲線を示さなかった。

ここで一度、乱巻きとして学習させた区間とその前後で、異常度（学習モデルの出力）に変化が生じるか確認を行った。結果は、図 1 2 に示すとおり、区間内では相対的に高い値を示すことが分かった。異常度は区間内では 0.8 から 0.9、区間外では 0.5 前後となっており、閾値を設けることで区別が可能に思われる。しかし、区間外で 0.9 付近まで異常度が上昇する場面も見られたため、閾値処理では誤動作が発生する可能性がある。

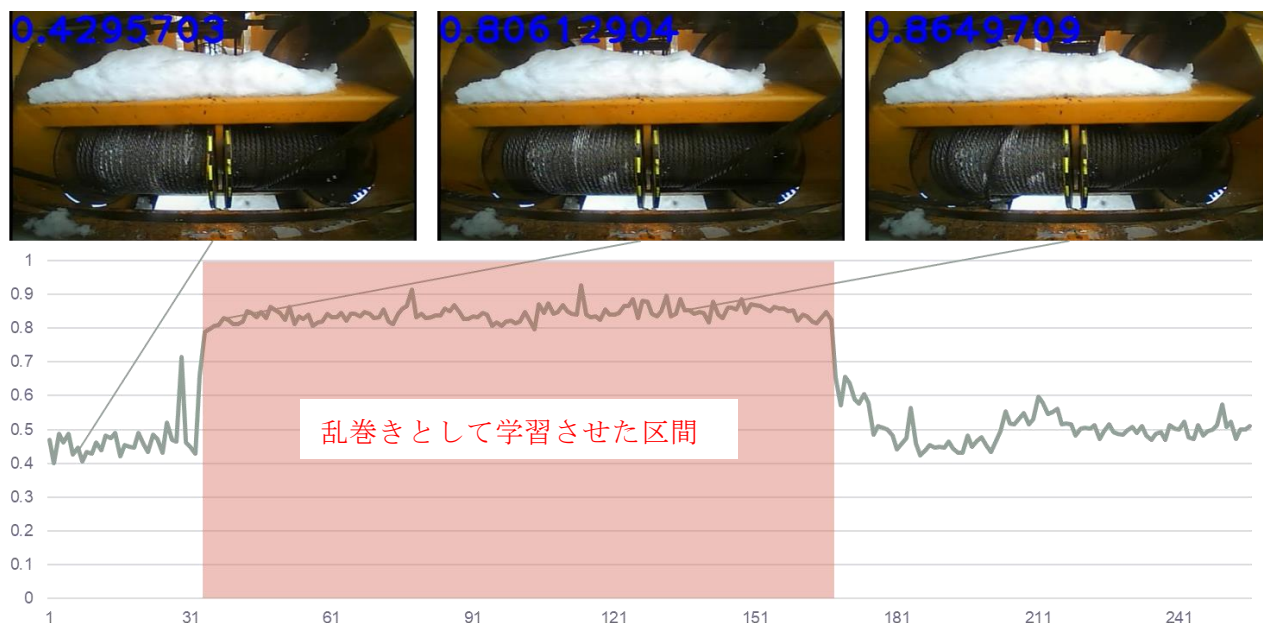


図 1 2 異常度の出力結果と画像フレーム

図 1 3 に認識結果を抜粋する。上段の正常と認識した画像と中段の異常と認識した画像は、ともに影が生じていながら正しく認識できた例である。下段の正常と認識した画像は、左側ウインチの左端に乱巻きが生じている、乱巻き未検知の例である。未検知となった例は、ウインチ全体が日陰に入っており、目視でもやや分かり辛い。

• 正常と認識した画像



• 異常と認識した画像



• 正常と認識した画像
(未検知)



図 1 3 認識結果（抜粋）

AI による手法において、学習モデルの精度はデータ次第であるため、精度が 100% となることはない。しかし、異常検知システムとしては、誤検知はある程度許容できるとしても、未検知は可能な限り減らしたい。そのため、後述する異常検知システムの出力部に工夫を施すこととした。

2. 4. 警報出力プログラムの作成

作成した学習モデルを組み込むことで動作する警報出力プログラムの作成を行った。概略を図 1 4 に示す。入力はクレーン車のカメラからの映像である。まず、映像をキャプチャし毎秒 1 枚の画像を出力するため、ライブ配信と録画が行えるツールである OBS Studio[6]を使用した。カメラ画像の取得試行時に使用した FFmpeg でも可能であったが、ファイル名に日付を付すことがより簡易に行えるため採用した。次に、学習モデルを組み込むための機械学習ライブラリである TensorFlow[7]を使用した。正確には、後述する Node-RED 上で動作可能な TensorFlow.js を採用した。最後に、データ処理のため、ビジュアルプログラミングツールである Node-RED[8]を使用し、異常発生時に音声出力が行われるようにした。

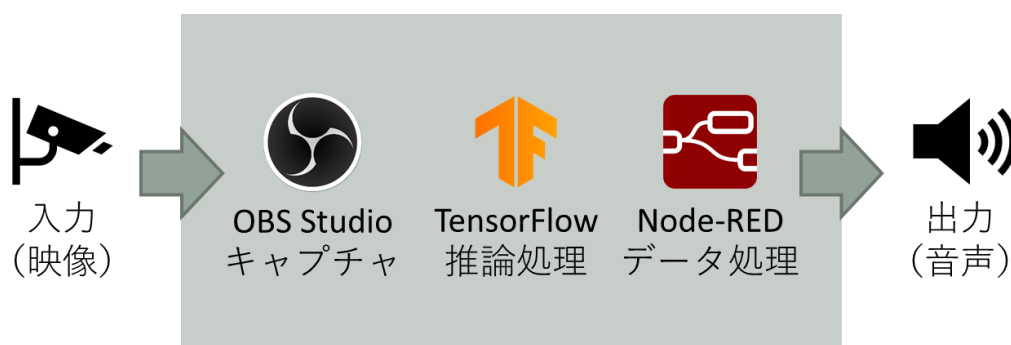


図 1 4 警報出力プログラム概略

Node-RED 上で行われる具体的な処理を図 1 5 に示す。まず、前段のキャプチャが出力したファイルが指定フォルダに追加されるため、その画像を読み込む。次に、画像を学習モデルに入力し、その出力として異常値を得る。最後に、それが閾値以上であった場合は異常とみなし、指定した音声ファイルを読み込み、音声出力を行う。

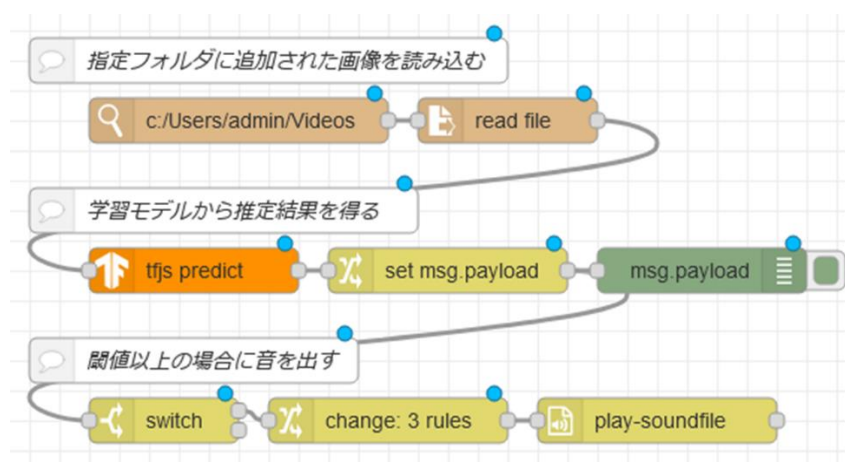


図 1 5 警報出力プログラムブロック図

2. 5. 異常検知システムの開発

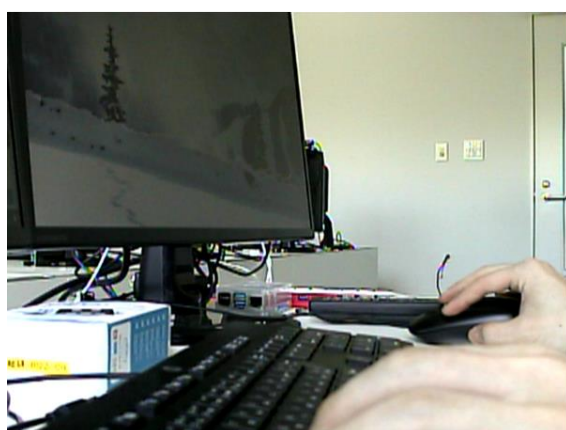
前節で作成した警報出力プログラムが動作する異常検知システムの開発を行った。選定したハードウェアは以下のとおり。

- ・ シングルボードコンピュータ (Raspberry Pi 4)
- ・ ビデオキャプチャ (Sanwa 400-MEDI008)
- ・ 外付けストレージ (256GB USB メモリ)
- ・ 車載充電器 (24V 対応 USB-C 5V/3A)

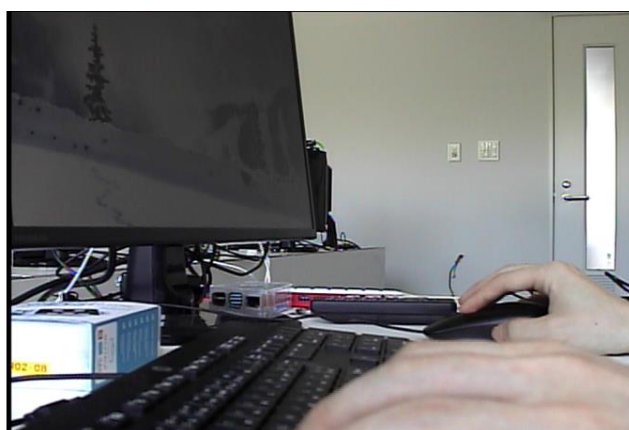
シングルボードコンピュータは、入手性が高く、安価で、画像データを扱うのに十分な性能を備えた Raspberry Pi 4 を選定した。ビデオキャプチャは、表 2 に示すように、入手性とシングルボードコンピュータ上での動作要件を満たすもののうち、画質の良好なものを比較した。画質の比較例を図 1 6 に示す。外光により白飛びしている領域が、左の画像より右の画像のほうが少ないことから、右のビデオキャプチャを選定した。外付けストレージは、1 日 8 時間駆動した場合に少なくとも 1 ヶ月間データを保存できる容量のものを選定した。車載充電器は、シングルボードコンピュータが要求する 5V/3A の電力を、クレーン車の 24V シガーソケットから給電するために必要となる。

表 2 キャプチャデバイスの選定

型番	入手性	Linux 上での動作
Princeton PCA-DAV2	×	○
Princeton PCA-DAV4	○	×
Buffalo PC-SDVD/U2G	○	×
IO-DATA GV-USB2	○	×
EasyCAP DC60+	○	○
Sanwa 400-MEDI008	○	○



EasyCAP DC60+



Sanwa 400-MEDI008

図 1 6 キャプチャデバイスの画質比較

その他、ネットワーク環境のない現場でファイル名に時刻を付与するための RTC モジュールをシングルボードコンピュータに搭載した。また、クレーン車のエンジン停止時に電源供給も止まるため、安全にシャットダウンしなくても動作に支障が生じないように、システム領域の ROM 化を行った。

最後に、所内で耐久試験を行い、データに欠損がどれくらい生じるかを確認した。図 1 7 は連続稼働時（約 3 日間）の書き込み回数の推移である。毎秒 1 枚の画像を取得するため、一時間あたり最大 3600 枚の画像が収集される。結果、平均欠損率は 2.4%（データの欠損が毎分 1 ～ 2 回程度発生）と、許容範囲内であった。また、長時間稼働させた場合も欠損率が上昇することなく、システムの耐久性に問題がないことを確認した。

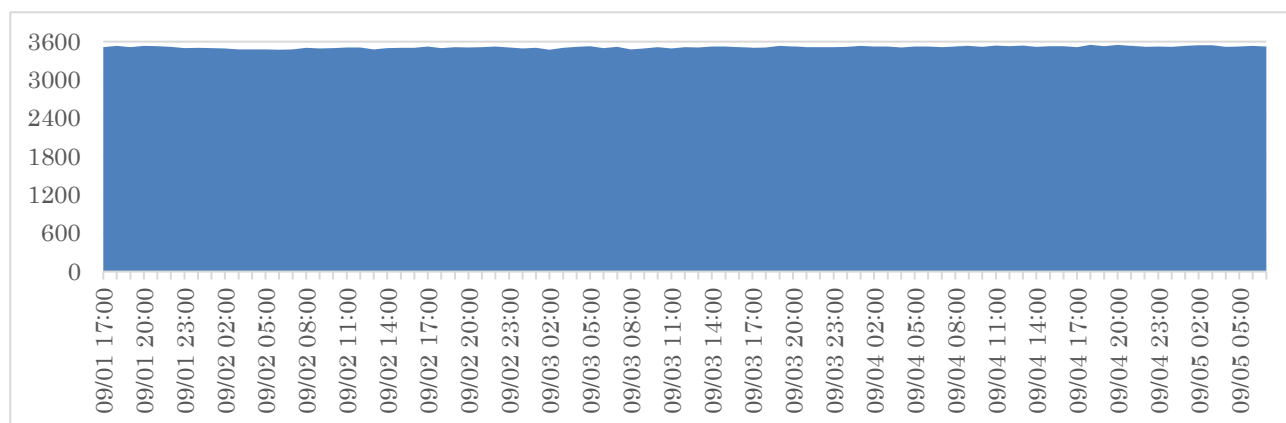


図 1 7 書き込み回数（回／時）

完成したデバイスの外観を図 1 8 に示す。筐体は、全体がヒートシンクの役割を果たすパッシブクーリングタイプのものを選定した。



図 1 8 異常検知デバイス

異常検知システムの出力部には、音声出力機能のほか、LED による通知を行えるよう、図 1 9 右に示すデバイスを追加した。理由は、実証実験時に警告音の誤動作によりオペレータの作業に支障が出ないよう、音声出力をオフにしたいとの要望があったためである。ここで、音声出力と同様、正常を緑色、異常を赤色で通知するのではなく、間に黄色（注意の意味）の LED を追加した。システム側に閾値を 2 つ設けることで、明らかに正常または異常の場合以外は黄色信号を通知されるようにした。これには、前節に記した精度の課題を、ユーザインタフェースで緩和しようという意図がある。

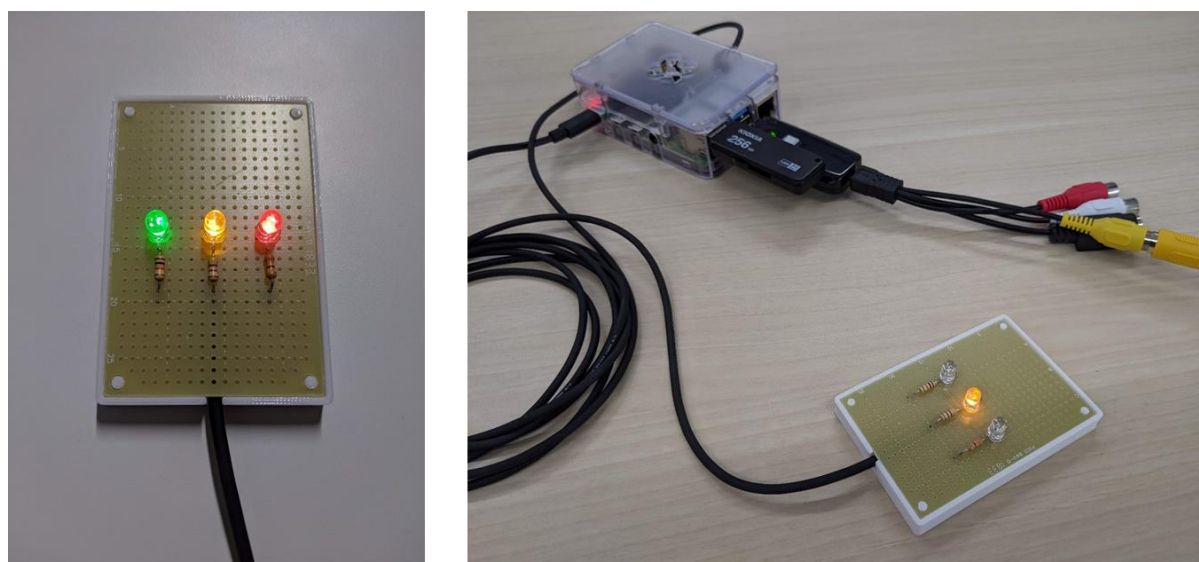


図 1 9 追加した出力装置

2. 6. カメラ非搭載のクレーン車への対応の検討

作成したシステムは、比較的乱巻きが起こりやすいラフタクレーンのうち一部の機種を対象としていた。しかし、システムを他機種に展開しようとした際、映像信号の取得方法が分からない機種があった。図 2 0 は、カメラ搭載クレーン車の映像出力端子が確認できた例であるが、端子が壁面裏にありアクセスが困難な機種や、どこに端子があるか不明な機種もあった。これらの機種に対応するため、カメラ付の異常検知システムを追加開発することとした。これにより、前述の機種のみならず、カメラが搭載されていない小型のクレーン車にも展開が可能となる。



図 2 0 カメラ搭載クレーン車の映像出力端子の確認

図 2 1 は、カメラへのアクセスができないクレーン車や、カメラ非搭載のクレーン車に、異常検知システムを搭載するための部材の検討内容である。カメラ本体は、ウインチ部上部に磁石で固定する想定で、防水タイプのネットワークカメラとした。また、カメラに対し電源供給も行う必要があるため、運転席のシガーソケットから、PoE（Power over Ethernet）対応のカメラへ、屋外用の PoE 対応 LAN ケーブルを運転席後方の開口部から通すこととした。間には、24V シガーソケットから電源供給可能な PoE ハブが必要となるため、新たに購入し、システム全体の動作確認を行った。



図 2 1 カメラ非搭載クレーン車に対応するための部材の検討

結果、下記の部材を揃えることで、カメラ非搭載のクレーン車にも対応可能な異常検知デバイスを構築できることを確認した。また、協力企業に依頼し、実際にクレーン車に搭載してもらったところ、カメラをしっかり固定するためにはより強力な磁石が必要であったため、市内のホームセンターで調達したとのことであった。後日に試験を行い、このシステムが問題なく動作することを確認した。

・部材（必須）

1. [シングルボードコンピュータ](#) ([microSD](#), [時計モジュール](#), [ケース](#))
2. [USB メモリ](#)
3. [PoE スプリッタ](#)
4. [PoE 対応 LAN ケーブル](#)
5. [産業用 PoE スイッチ](#)
6. [ヒューズ内蔵シガープラグ](#)
7. [PoE 対応屋外用 LAN ケーブル](#)
8. [ネットワークカメラ](#)

・部材（オプション）

- ・ ノート PC と LAN ケーブル（カメラの向きを現場で調整する場合）
- ・ スピーカー（異常発生時に警告音を出力する場合）
- ・ ビデオキャプチャと映像取り出しケーブル（既設のカメラから画像を取得する場合）

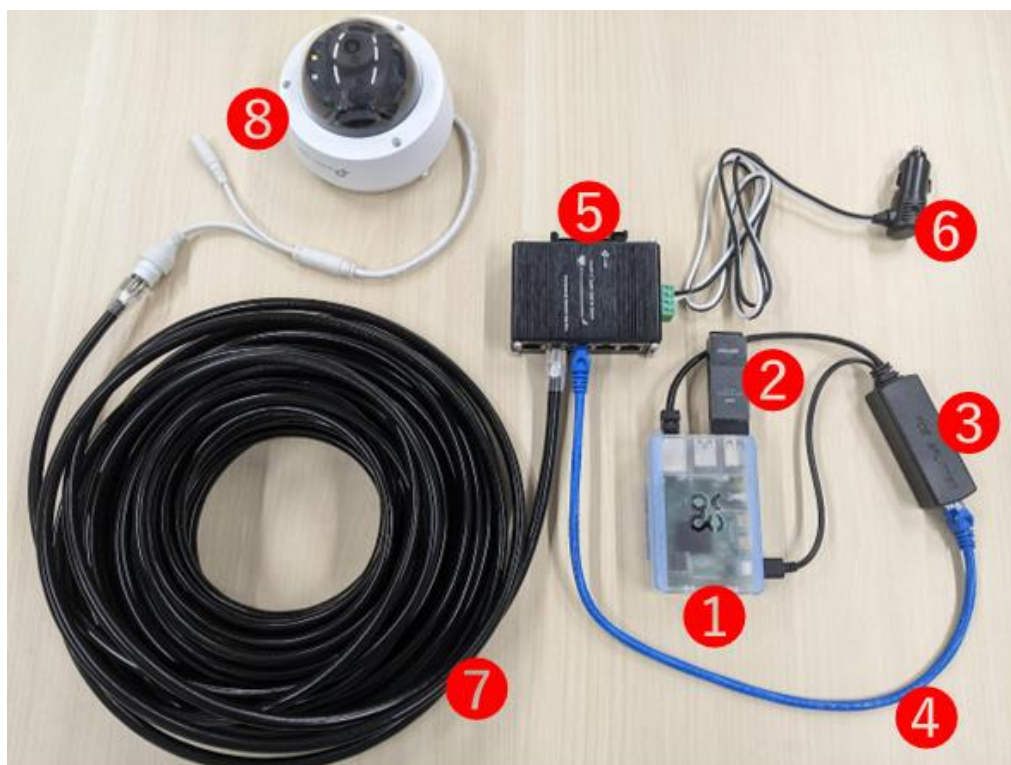


図 2 2 カメラ非搭載クレーン車への対応のための部材一式

2. 7. 実証実験

実験にあたり、これまでに収集したデータを学習させた。収集した画像のうち、一部欠損のある画像や、著しく不鮮明な画像を取り除いた上で、正常画像については1分間に1枚程度に間引いた。

そのため、最終的な学習データは、乱巻き画像約2000枚を含む約60000枚のウインチ部画像となった。アルゴリズムは試行時と同様、CNN(Convolutional Neural Network)を用いた。また、評価結果として正解率を算出した。

図2-3に学習曲線を示す。試行時と比較すると、Training Error (赤実線)、Validation Error (赤点線) とともに低い値で収束しており、良く学習できていることが分かる。

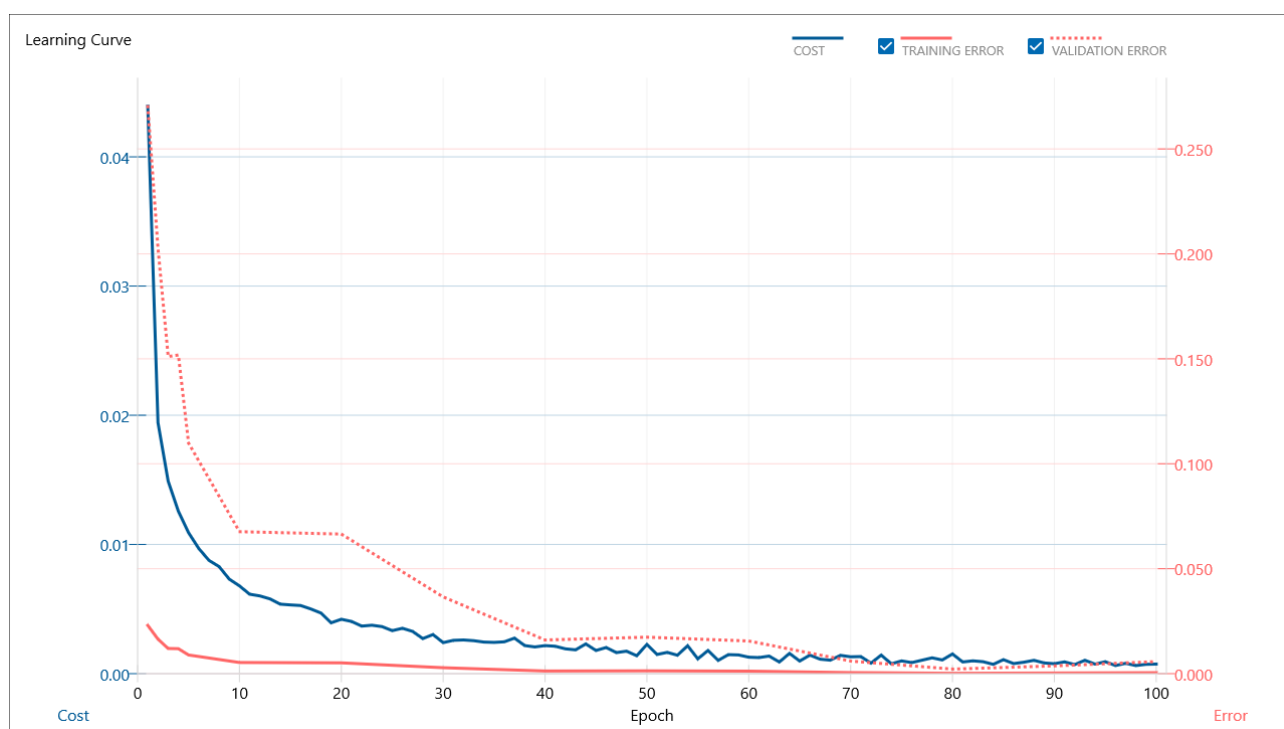


図 2-3 学習曲線

また、学習データとは別に用意した推論用データを用いて、推論を行った結果を図2-4に示す。横軸は出力値、縦軸はデータ数である。また、赤色は試行時の結果、青色は今回の結果である。正常を0、異常を1として学習させているため、両端にデータが集まっているほど良く分類できているといえる。試行時は0.8付近にピークがありつつ全体に分散しており、正常と異常を判別できていない様子であったが、今回は0と1にピークがあり、試行時と比べてよりはっきり判別が行われていることが分かる。

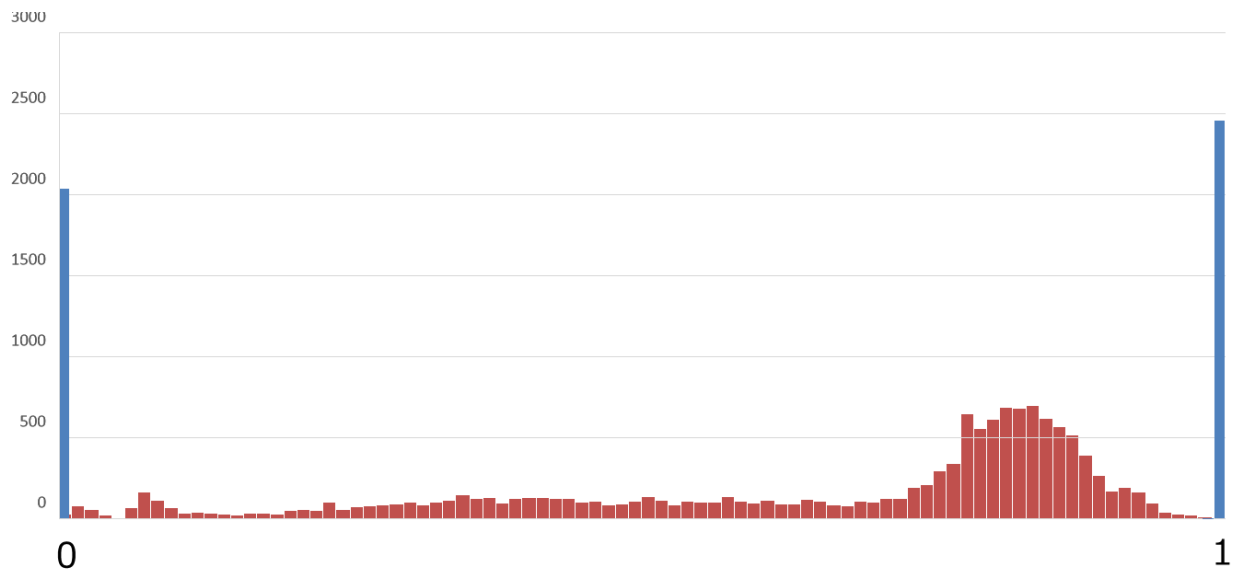


図 2 4 推論結果のヒストグラム

図 2 5 に示すとおり、正解率は約 99.3%となった。また、画像データ 4541 枚中、未検知は 29 枚、誤検知は 1 枚となった。

	y'=0	y'=1	Recall
y:label=0	2031	1	0.9995
y:label=1	29	2480	0.9884
Precision	0.9859	0.9995	
F-Measures	0.9926	0.9939	
Accuracy	0.9933		
Avg.Precision	0.9927		
Avg.Recall	0.9939		
Avg.F-Measures	0.9932		

図 2 5 推論結果 (Neural Network Console の出力画面)

実証実験では、今回作成した学習モデルを組み込んだ異常検知デバイスをクレーン車に搭載し、意図的に乱巻きを発生させ、得られた推論結果から正解率を算出した。実証実験の様子を図 2 6 に示す。クレーンのブームを下げることで、ウインチ部のワイヤロープが緩み、乱巻きが発生しやすくなる。



図 2 6 実証実験の様子（ブームを下げているところ）

図 2 7 は、クレーン車ウインチ上部の既設カメラと、運転席左側に設置した異常検知デバイスである。LED 搭載の出力装置は実証実験用に仮設置したもので、本来は着席時に見つらい位置である。



図 2 7 左：クレーン車に既設のカメラ 右：異常検知デバイスの搭載箇所

実証実験の間、乱巻き時も正常の LED が点灯している様子であった。結果とその考察は次項に記す。

3. 実験結果

実証実験の結果は表3のとおり。学習モデルでの評価実験では正解率 99.3%、クレーン車での実証実験での正解率は 60.6%となった。

表3 各実験結果

学習モデルの評価実験での正解率	99.3%
クレーン車での実証実験での正解率	60.6%

4. まとめ

学習モデルの評価実験では、乱巻きの検知が精度よく行えることを確認した。クレーン車での実証実験での正解率が相対的に低い理由として、学習データ取得時と実証実験時との気象条件が大きく異なっていたことが考えられる。図28に示すとおり、学習済みの画像には積雪が見られ、実証実験時の画像には見られない。また、実証実験時の画像はレンズの汚れのせいか写りがやや不鮮明のように見える。このように、学習したデータと推論データの条件が大きく異なるため、正解率が低下したものと考えられる。ちなみに、乱巻き画像は日程の都合上全て同日に撮影されたもののため、条件の異なる場面での乱巻き画像を学習させることでより頑健になると思われる。



図28 学習済みの画像（左）と実証実験時の画像（右）の比較

5. 参考文献

- [1] 地方独立行政法人青森県産業技術センター，株式会社N I C H I U N（2022）“ワイヤロープ異常検知システム、方法、ウインチ及びクレーン” 特開 2022-144522.
- [2] クレーンオペさんのブログ <https://beherenow982.com/158/>
- [3] Teachable Machine <https://teachablemachine.withgoogle.com/>
- [4] MathWorks <https://jp.mathworks.com/discovery/autoencoder.html>
- [5] Neural Network Console <https://dl.sony.com/>
- [6] OBS Studio <https://obsproject.com/>
- [7] TensorFlow <https://www.tensorflow.org/>
- [8] Node-RED <https://nodered.org/>