東京工業大学情報理工学院総合型選抜　活動実績報告書

氏名 (ふりがな): 前田 恭輝 ( まえだ　たかき )

高等学校: 愛知県立愛知総合工科高等学校   
(2018年3月 卒業・卒業見込み)

活動実績概要 (150字程度):

会社で3年間の設備保全業務を通して、感じた課題が多くあり、ITを用いて解決できるようになりたいと思った。そこで、ITについて資格を通して学び、実際にアプリを考えて、学習しながら実装を行った。  
今回は、今後進みたい分野である機械学習も実際に使ってみて、様々な知見を深めることができた。

活動実績の実施状況:

志願者が単独で行った

教師などからの指導を受けながら志願者が単独で行った

共同で行った

その他

報告書本体ページ数(表紙を含まない): 4ページ

注意:

* 報告書本体を4ページ以内で作成し、この表紙と一緒に提出すること。
* 報告書本体の形式は自由とするが、文字の大きさは10ポイント以上にすること。また内容として活動実績の背景、具体的な内容、活動実績の実施状況の説明、参考にした資料の一覧などを必ず含むこと。
* 報告書本体に、活動実績を志願者が単独で行ったか否か、共同で行った場合は自身の役割、指導を受けた場合はどの部分に対する指導か等の説明を書くこと。
* 報告書本体には氏名、学校名はどうしても必要な場合を除いて書かないこと。

機械学習を用いた動画像分析による

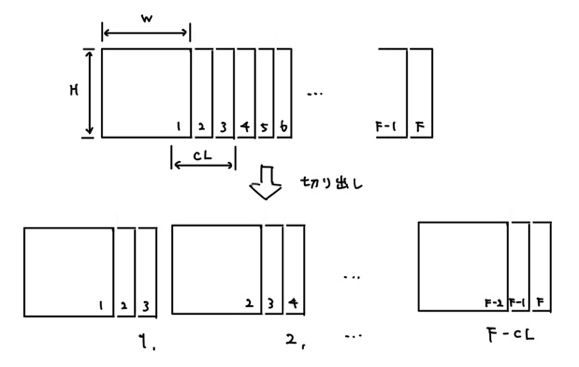
異常検知アプリケーションの実装

１．背景と活動概要

　私は、トヨタの製鉄会社にて主に大型クレーンの保全に携わってきて、多くの課題を感じていた。そのうちの一つに、点検時の巻き込まれや転落等の安全性と点検項目の多さがある。そんな中、AIの発展で自動化がより加速し労働者が減っていく一方で、自動化していく機械は変わらず人がメンテナンスをしていく必要がある。それどころか、自動化が進み、安全、品質基準が上がるにつれて、点検項目は増えていく。実際に働いていた現場では、人手不足も深刻化していて、点検が十分に行えていない現状があった。

　そこで、私は保全業務を支援できるようなシステムやアプリケーションの提案をしたいと考え、ITについて資格を通して学び、近年進化する機械学習を活用できないかと思い至った。中でも、点検頻度が高く、巻き込まれ等のリスクがあり、トラブルの起きやすい駆動部の点検にフォーカスした保全の支援を考えた。具体的には、動画像を機械学習手法を活用して分析し、正常時との比較を行って異常を検知できないかと考えた。

　また、各フレームを特定長に切り出したものを評価する（図１）ことで、既存の機械学習を用いた画像分析に加え、正常時より速い、遅い、異なる“動作”を検知できるアプリケーションを考えた。そして、Pythonと機械学習を学習しながら、アプリの実装を試みた。

 (図１)

（H：Height、W：Width、F：総フレーム

CL：切り出す長さ）

２．活動内容

2.1 環境構築

動作環境：MacBook Pro 13inc

使用言語：python

使用ライブラリ：Numpy（演算処理用）、

OpenCV（動画像処理）、Matplotlib（グラフ描画）、

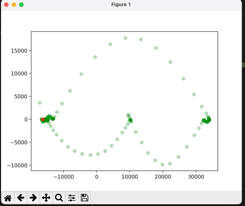
scikit-learn、Keras、TensorFlow（機械学習）

上記の環境で行った。

2.2 プロトタイプの設計と実装

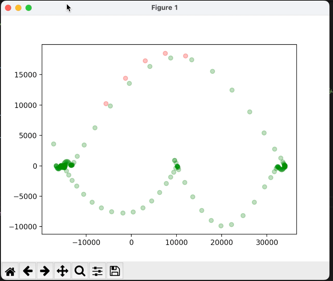
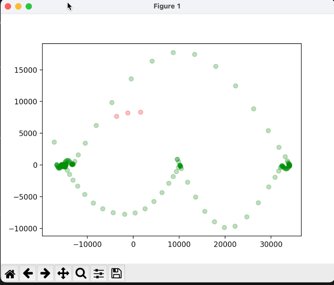
まず、動画像を解析する上でそのまま扱うと莫大なデータ量になってしまうので、次元削減と言われる手法を用いる。これは、データのもつ特徴をより低次元なデータで表す手法で、今回はと呼ばれる次元削減手法を用いる。これにより、大幅に計算コストを削減することができる。

次に、これを特定長のフレームに切り出すクリップ処理(図１)を行う。この、連続するフレームに対し、更に2次元へ次元削減する事で、正常動作のクリップが座標平面上でどのように分布するかを調べ、それにそぐわないデータを異常データとみなす事で、異常な動作を検知できないかと考えた。ここで、異常データを分類する手法として、シンプルなと呼ばれる手法を用いることにする。以下、一回目での次元削減を一次、2回目での次元削減を二次と記す

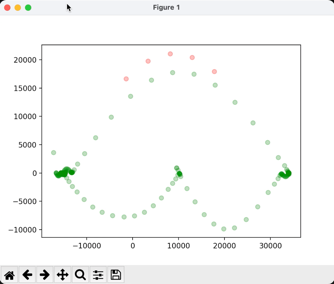
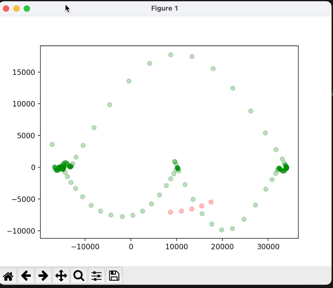
　まず、正常データの各フレームを2次元に次元削減した後、クリップ処理を行って、これを更に2次元へ次元削減を行った。すると、図のように分布した。　　 正常データの分布

（検証データは物体を画面内の３点で手動移動させる、簡単なものを用いた。）

ここで、正常動作と比較し、遅い、速い、異なる動作をしたところ、確かにデータが正常データに対し分離できていることが確認できた。（緑点が正常なデータ、赤点がリアルタイムでプロットしているデータ）

正常動作 　遅い動作

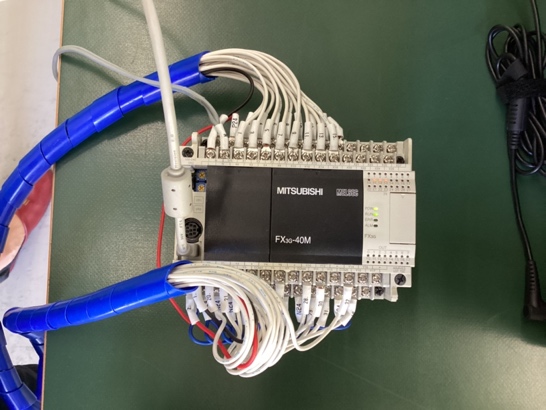
速い動作 　　　　異なる動作

次に、k近傍法分類するために異常データをランダムにプロットした点からn番目に近い正常データまでの距離が閾値を超える場合、その点を異常データとして配列に格納するという方法で作成した。

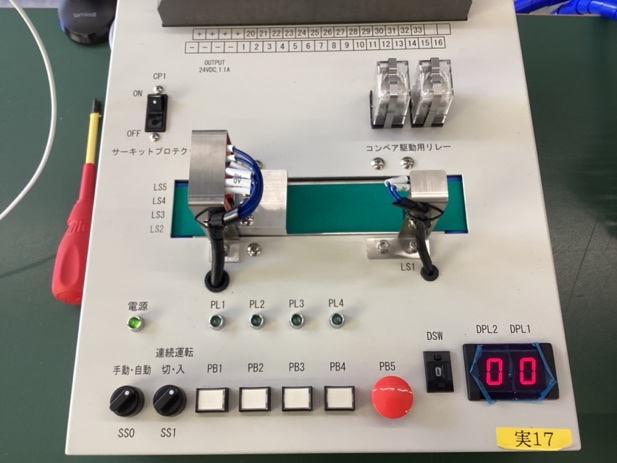
最後に、k近傍法で分類する為に、正常データと異常データにそれぞれラベリングをした。

2.4 テスト

　シーケンサで制御できる装置を用いて、実際の機械のような精密な動きで異常を検知できるかテストした。

　　　　撮影風景　　　　　　　　シーケンサ

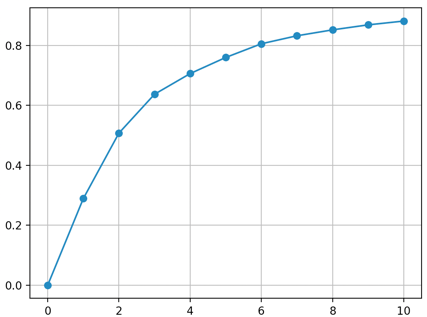
すると、明らかな速度異常や動作異常でないと検知できないことがわかった。また、ランプの異常点灯や指の写り込み等の異常が検知できないことがわかった。　　テスト装置と制御する動作

ランプ点灯

往復動作

加えて、正常データでピントズレが起きた際、分布が大幅に変化して、正しく異常検知できなかった。

2.5 原因調査

まず、次元削減による情報ロスが考えられたので、削減後の次元数に応じたデータの復元度を測定した所、図のような結果であった。(縦軸は復元度、横軸は次元数)

これによると、２次元への次元削減では半分程度の情報量しか保持できておらず、ランプ等の小さな画素変化が表現しきれていないことが考えられた。

また、引用[3]より、主成分分析は学習させてない画像を学習させた画像に復元しやすいことがわかった。これにより、一次の次元削減で異常な画像でも正常な画像のように表現し、二次の次元削減で正常データとの区別がつかなくなっていたと考えられる。

　加えて、分散が最大になるように主成分を合成することから、正常データに正常時から大幅にずれた(ピントズレ等)ノイズが含まれると、そのデータを表現するために正常時のデータの分布が大幅に圧縮され、正常時付近の異常データの検知ができなくなっていたと考えられる。

2.6 対策

　まず、削減後の次元数をデータの復元度が90%以上となるように、次元数を設定する。その際、距離を用いた分類手法である現手法は、球面集中現象と呼ばれる、次元数が増えるにつれて球の体積が表面に集中していく現象によってうまく分類できなくなることがわかった。その為、これとは異なる手法を用いる必要がある。加えて、現手法のような教師あり学習は異常データが確保できているときに有効な手法であり、今回の方法は本質的には教師なし学習で、無理やり教師あり学習の手法を適用している形であることがわかった。

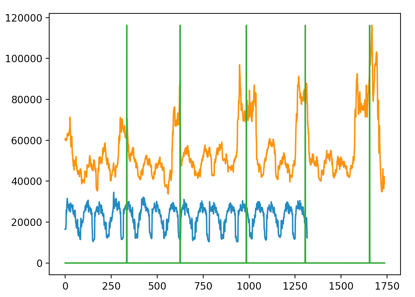
　次に、主成分分析では学習していないデータを次元削減すると、正しく復元できない。この事を利用し、復元前後で差分を取ることで異常度を測る教師なし学習の手法があり、その手法であれば、ランプや指の写り込み等の異常検知も期待できることがわかった。

　以上のことから、主成分分析を用いた教師なし学習による異常検知モデルを考えることにした。

2.7 アプリケーション改良案と実装

試行錯誤の末、図のようなモデルを考えた。ここで、削減する次元数は復元度が90%以上になるように次元数を設定した。また、異常度は復元前後の差分の絶対値の全画素の総和によって評価する。

すると、正常データ（青）と異常データ（橙）は以下のようになった。ここで、縦軸は異常度、横軸が時間となっており、異常データの異常モードは左から、正常、高速、低速、ランプ点灯異常（異常画像）、ランプ点滅異常（異常動作）となっていて、緑線によってモードの区切りを示している。また、異常画像と異常動作は異常な画素が小さく（全体の1.96%）、速度の異常度は小さい（正常時の+6.3 %と-15.0%の速度）ものを使った。



　　　一次異常検知

想定では、一次では異常画像のみ、二次では異常画像以外の異常モードで異常を示すはずであるが、この結果によると、全体的に異常度を高く示しており、一次では異常画像に加えモードの切り替わりで異常を示していた。

2.8 原因調査

　ここで、異常度を示すデータを画像に復元し確認してみると（図２）、ベルトコンベアの繋ぎ目が正常時と異なる位置にあり、常にベルトコンベア全体が異常を検知していたことがわかった。これにより、二次で異常度が正常時も高く出ていたと考えられる。また、制御装置の操作時に自分の手によって影が変化していることがわかった。これにより、各モードの切り替わりで異常度が上がっていたと考えられる。

2.9 対策

前者は、異常度の映像にあるように、各々の画素の異常度自体は大きくないが、占める画素が広範囲であるために、総和で算出した異常度に大きく影響していると考えられる。その為、異常画素の異常度は大きいものとして考えれば、異常度が閾値以下のものを無視して算出することで、実際の異常のみを抽出できると思われる。

　以上より、異常度のフィルタリングを行った結果以下のような結果を得ることができ、確かに異常を検知することができた。

　しかしながら実用的には異常度の分離は不十分であり、環境光や画角のズレ等に対しても影響を受けてしまう事から、これらに対し適切な前処理が必要である。

2.11プロトタイプの改良＋α

　ここで、主成分分析以外の機械学習手法を用いたモデルを考えて、実装、比較してみた。今回は、自己符号化器（AutoEncoder）を実装してみた。理由として、ノイズに強く、主成分分析の時に大きく影響していた一瞬の大きい画素変化（ピントずれ等）に対し強くなることが期待できることと、ニューラルネットワークによる機械学習手法で、知見を深めることができると考えたからである。AEのモデルは以下のように実装した。

異常度を算出してみた所、図のように精度の向上が確認できた。

３．結果

表.1

４．まとめ

今回の活動は、課題解決の為にアプリケーション案を考案し、実際に機能するかをプロトタイプを通して確認した後、異なる機械学習手法を用いて改良を試みた。

結果として、人間が見てわかるような、明らかな異常に対しては異常が検知できたが、わずかな速度変化や異常画像に対しては十分な精度が確保できなかった。また重大な欠点として、画角がずれてしまうと、異常検知に影響が出てしまうという点がある。これに対し対策が必要であるが、具体的な解決策が思い付かないので、今後の学習を通して解決策を模索していきたい。

また、このアプリケーションを実用段階にするには、異常を知らせるアラーム機能、異常が検知されたクリップ映像を書き出す機能、異常領域の示唆、また、それが正常であったときの追加学習機能やアプリケーションのUI実装、デバイス化やIoT対応等の課題が考えられる。

加えて、今回はフーリエ解析等の知識不足によって断念したが、音声分析による異常検知等もぜひ実装したいと考えているので、今後の展望としてこれらの実装へ向けより一層学習を進めていきたいと考えている。

最後に、この活動を通し、pythonプログラミングや各機械学習手法について知見を深めることが出来た。特にpythonプログラミングに関し全くの無知であった活動当初と比べとても成長できたと感じているので、今後の学習や活動にもこれらの経験をぜひ活かしていきたいと思った。

参考文献

[1]主成分分析<https://www.iplab.cs.tsukuba.ac.jp/paper/master/nogami_master.pdf>

[2]k近傍法<http://www.shalab.phys.waseda.ac.jp/pub/pdf/mt1.pdf>

[3]<https://statistics.co.jp/reference/software_R/statR_9_principal.pdf>

[4]<https://www.youtube.com/watch?v=4Vk1UhRDB34&t=2021s>

[5] その他pythonに関する多数記事