

経常研究

データ収集・解析システムの機能拡張

ー データ収集・解析システムユーザインタフェースの拡張 ー

島田 智* 八木澤 秀人*

Enhancements in the Data Collection / Analysis System
SHIMADA Satoshi and YAGISAWA Hideto

当センターがこれまでに開発してきた、データ収集・解析システムの見える化部分について、表示内容のカスタマイズをブラウザ上から簡単にできるよう、機能の拡張を行った。また、ブラウザ上からの設定で、7セグメント表示器やアナログメータ読み値の教師データを作成するツールや、クラウド上で推論モデルを生成するツール、推論モデルを用いて、画像から値を読み取り、データを保存するツールを作成し、実際にセンターの消費電力計やアナログメータの読み取りを行うモデルを構築、継続的に値を収集し、これらのツールの有効性を確認した。

Key words: IoT, 可視化, YOLO, アノテーション

1 はじめに

ものづくり企業を取り巻く環境は、新興国の技術革新に加え、コロナ禍による半導体不足や原料高騰により、生産性向上、競争力の強化が求められている。顧客満足度を高めるためにも、機会を捉え、リスクを回避するためにも、データに基づく素早く適切な判断が必要で、IoTやAI等の先端技術利用が不可欠である。しかし、ものづくり現場においては、作業日報や品質情報などのデータ収集を行っているものの、データ収集の電子化・自動化や、収集したデータの活用が進んでいない企業も多い。

そこで産業技術センターでは、平成30年の「所内設備のIoT化に関する研究¹⁾」から3年間にわたり、ものづくり現場で発生する様々なデータの利活用を目的として、データの収集、蓄積、分析を行うためのツールを開発^{2),3)}を行ってきた。

こうしたツールを予め準備し、利活用することで、研究会でのハンズオン活動がスムーズに実施できるとともに、相談を寄せられる多種多様なIoT、AI利活用に関する企業ニーズに素早く対処することが可能になっている。

一方、研究会参加者の多くが、システム管理やプログラミングの経験に乏しく、こうしたツールを企業内で使い続けることを考えた場合、長期的な人材育成が不可欠である。また、多品種少量生産を行う企業にとって、システムを一度作り込んでも、表示するデータや、その見

せ方の変更が必要となることが想定される。

そこで本研究では、ユーザが自ら望んだ形にシステムのカスタマイズを行えるよう、これまでに構築したデータ収集・解析システムの機能の拡張を行った。

2 研究の方法

2.1 見える化ツールの機能向上

これまでの研究では、データ収集・解析システムにRaspberry Pi 4B上のApacheで動作するオリジナルのPHP見える化プログラムを作成した。このプログラムでは、ダッシュボード(図1)(複数の情報源からデータを集め、概要をまとめて一覧表示する画面)の表示内容を、JSON形式のテキストファイル(表1)で管理している。このため、希望する表示対象データを選択し、グラフの種類、キャプション等の表示方法を得るには、このファイルの編集方法、編集箇所、記載内容について、細かな内容まで理解する必要がある、ユーザが容易に変更できるものではなかった。

こうしたことから、以下の2つの手法により、ユーザが容易にデータ表示画面をカスタマイズできるよう、機能の拡張を行った。

2.1.1 ダッシュボード編集のためのUI拡張

まずは、現在のデータ収集・解析システムのトップページである見える化ツールのダッシュボードを、テキストファイルの編集をすることなく、ブラウザ上から、ボタンや選択リストで編集できる機能を加えた。

* 栃木県産業技術センター 機械電子技術部



図1 ダッシュボード画面
表1 パネル毎の設定内容

```
{
  "title": "eCO2 濃度",
  "text": ". / dashboard.php?MODE=FULL
  &GW_ID=3&DEV_NO=1&CH_NO=5&BG=LG&UNIT=PPM
  &NumDEC=0&NumMIN=400&NumMAX=1000&COUNT=120
  &COLOR=255, 200, 200&THICKNESS=5",
  "link": ". / chartview.php#dt5",
  "update": "100"
}
```

2. 1. 2 統合監視ツールやBI ツールの活用

もう一つが、統合監視ツールやBI ツール等、外部のツールを活用する手法の検討である。こうしたツールの中には、FOSS（フリーソフトウェアやオープンソースソフトウェア）や、機能が限定された範囲ならば無償で使うことのできるものが存在している。独自ツールと比較すると、表示手法や連携可能なデータベース等の機能、構築や操作、トラブル発生時の対処方法等に関するマニュアル等の情報が豊富で、運用のハードルが低い（表2）。そこで、こうしたツールの中からいくつかを選び、収集したデータの表示を試み、独自ツールの置き換えを検討した。

表2 ツールの機能比較

ツール 比較項目	オリジナル 見える化ツール	統合監視ツール	BI ツール
費用	○	△ 多くが有償	△ 多くが有償
必要な使用スキル	△	○	○
導入・管理スキル	△	○	○
セキュリティ・サービスの持続	△ 独自に維持していく必要	○ ベンダー側で対応	○ ベンダー側で対応
カスタマイズ	○ 細かな作り込みも可能	△	△

2. 2 計器読取プログラムの機能向上

データ出力が難しい加工機からデータを収集するには、振動、電流、電圧、リレーやマグネットスイッチの開閉などをセンサで監視するほかに、装置に備付けの計器の画像を用いる方法がある。特に 7 セグメント表示器やアナログメータは、ヒトが目視で状態を監視するために多くの装置に付けられている。また、これらの装置の状態を連続して記録するために、様々な製品や安価に構築する手法が開発されている。

これらは大別すると、パターンマッチや輝度の差などをベースとした手法と、物体検出アルゴリズムなどのDNNを用いた手法とに分けられる。

前者は後者に比べると計算量が少なく済み、高速で安定している一方、環境に応じたパラメータの設定が必要となし、明るさの変化や外光の映り込み、カメラのズレなどの外乱には比較的弱く、うまく認識ができなくなる恐れがある。一方後者は一度外乱に強いモデルを構築すれば、カメラのズレや明るさの変化にも対応できる一方、計算量、計算時間は余計にかかることになる。

これまでの研究では、物体検出モデルである YOLOv3 を用いた 7 セグメント表示器の読取りモデルを構築し、電力値の読取りを行ってきた。そこで、今回は実際の企業での活用を考え、YOLOv3 よりもサイズが小さくても高い精度が出るとされる YOLOv5 を用いて、アノテーション作業（図 2）からモデルの構築、活用までの環境構築を行った。

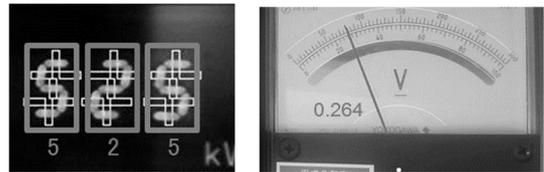


図2 アノテーション作業

画像にラベル（今回の場合は読み値）付けしていく作業

2. 2. 1 モデル作成環境の構築

7 セグメント表示器やアナログメータについて、様々な画像を学習させることで様々な計器に対応したモデルを作ることも可能である。一方で、対象を特定の計器に絞ることで、より安定した読取りを実現できる。この場合、対象物毎に、読取りのためのモデルを構築する必要がある。

今回作成するモデルは、画像を入力として、読み値を返すもので、学習には大量の画像とその読み値を与える必要がある。画像に読み値を与えていくアノテーション作業は、時間以上に苦痛を伴う作業であることから、まず、従来手法により画像と読取り結果を同時に収集するプログラムを作成することで、アノテーション作業の軽減を試みた。

また、このプログラムにより得られる画像とラベルを教師データとして、転移学習により、画像から読み値を推定するモデルを生成するツール、構築したモデルを用いて、画像から値を読み取り、csv 形式のファイルや DB にデータ保存するツールを作成した。

2. 2. 2 作成したモデルの実タスクへの適用

センターの消費電力量表示と、太陽光パネルからニッケル水素電池に充電を行う回路に取り付けたアナログ電圧計、電流計を用いて、実際にモデルを構築し、アノテーションに用いた従来手法との比較を行った。

3 結果及び考察

3.1 見える化ツールの機能向上

3.1.1 ダッシュボード編集のための UI 拡張

まずはダッシュボード上のパネルダッシュボード一つの表示につき、対象となるデータや背景のグラフ表示の方法について、選択式で表示、更新するプログラム(図3)を PHP で作成し、加えて JavaScript を用いて、画面表示を選択内容に合わせて変更し、設定内容を自動生成するプログラムを作成した。しかしながら、前述のとおり、見える化部分を自作し、機能を拡張していくことには、課題が多い。

図3 作製したダッシュボード編集画面

3.1.2 統合監視ツールや BI ツールの活用

そこで今回、BI ツールや統合監視ツールとして、無償で使用できること、導入や使用方法、導入事例についての情報が豊富なことなどを考慮し、Google Data Portal⁴⁾、Microsoft Power BI Desktop⁵⁾、Grafana⁶⁾、Zabbix⁷⁾を用いて、見える化ツールの置き換え可能性の検討を行った。

3.1.2.1 Google Data Portal

Google Data Portal は、Google が提供する、基本料無料のクラウド BI ツールであり、Google BigQuery、Cloud SQL、スプレッドシート等のデータが利用可能である。Google のアカウント情報を基にデータの共有、編集権限の割り当てなどが可能で、レポート作成機能に特化(日報、月報等の作成が得意)している。

データ収集・解析システムのデータは、MariaDB 内に格納されており、Google のサーバからアクセス可能な場所にこの DB を配置することで、データの連携が可能である。

実際にデータの表示を行ってみたところ、日報等のレポート作成には非常に有効なツールである一方、定期更新が必要なリアルタイム監視は不得手な印象がある。特に今回、DB に Google 内のサービスを使わなかった為か、1 か月分の温湿度データの表示に 1 分以上要するなど、少し大きめのデータを参照しただけで、非常に大きな遅延が発生し、見える化ツールの置き換えには不向きと判断した。

3.1.2.2 Microsoft Power BI Desktop

Power BI Desktop は、Microsoft が提供する無償版

を有する BI ツールで、PostgreSQL、MariaDB、excel 等、様々なデータベースが利用可能である。デスクトップ版と携帯版のページを別に作成することが可能で、データ表示範囲の拡大、縮小による傾向の把握が行いやすい。また、報告書を「発行」することで、出先でも確認できるが、複数ユーザでの共有には Pro 版(有償)が必要となる。

PC から LAN 内部にあるサーバのデータを参照するのであればそこまで遅延が大きいとは言えないが、やはりリアルタイムでの監視に使用するには不向きと判断した。

3.1.2.3 Grafana

Grafana は、マルチプラットフォームで動作する視覚化ツールで、PostgreSQL、MariaDB 等、様々なデータベースが利用可能である。様々なサービスの可視化に活用されており、ダッシュボードの表示内容をパネル毎に設定していき、ブラウザ上でその配置や大きさを変更することが可能である。また、ユーザ毎にパスワードを設定し、画面毎に編集や閲覧の権限を管理することができる。

一度システムにインストールをしてしまえば、表示するデータの選択から表示方法の変更、データのエクスポートに至るまで、全てブラウザ上で行うことができる、非常に強力なツールであり、見える化ツールの置き換えに最も適していると考えられる。一方、通知機能も簡易ながら有しているものの、単体で状態監視を行うには機能の不足が感じられる。

3.1.2.4 Zabbix

Zabbix は、世界中で使用されているネットワーク機器の統合監視ツールであり、Zabbix サーバと監視対象に導入する Zabbix エージェントから構成される。Zabbix サーバは指定した時間毎に監視対象のエージェントにデータを要求し取得する。また、WebUI を有し、ブラウザから各種設定が可能で、データの可視化機能も有している。通知の機能が豊富で、データの扱い(保存期間等)も設定できることから、長期間運用する場合に有用である。Zabbix と Grafana を用いた場合のシステム構成図を図4に、作製したダッシュボードを図5に示す。

また、Zabbix Agent2 を導入することで、既存システムが受信するセンサデータを、MQTT Subscriber として受け取ることができ、Python で作成した MQTT で受け取ったデータを MariaDB に保存するプログラムを用いる必要がなくなる。

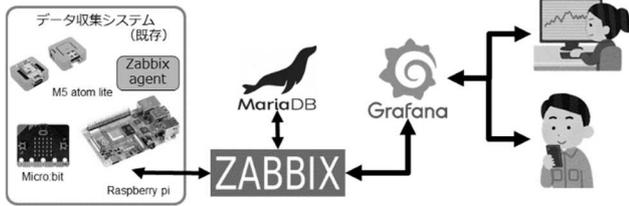


図4 システム構成図



図5 Grafanaを用いたダッシュボード

3. 2 計器読取プログラムの機能向上

3. 2. 1 モデル作成環境の構築

今回、読取り対象とする計器は、7セグメント表示器とアナログメータであり、転移学習のベースモデルにYOLOv5n⁸⁾、VGG19⁹⁾を用いた。

今回のタスクにおける教師データ作成(アノテーション作業)は、画像を見て、その読み値をテキストに保存する単純作業ではあるが、数が多くなると容易ではない。実際に作業を行ってみると、3桁の7セグメント表示器画像にアノテーションツールlabelimg¹⁰⁾を用いてラベルを振っていく作業には30秒/枚程度、画像ファイル一覧を記載したテキストファイルにアナログメータ画像に読み値をラベル付けしていく作業には20秒/枚程度を要した。今回の場合、位置が固定なので、7セグメント表示器の画像についてもlabelimgを使わず、テキストで次々に値を記載し、後で画像毎のデータに変換することで時間を短縮できるが、すべての画像を目視し、その値を入力する作業を計器毎に100枚、1000枚と行うのは、作業時間以上に苦痛を伴う。

そこで、7セグメント表示器については、各セルと周囲との輝度差を用いて、アナログメータについては、針位置をハフ変換により抽出する手法を用いて読み取るプログラムを構築した。

まず、これらの手法での読取りに必要な表示器を映すカメラのアドレス、7セグメント表示器各桁のセルの位置、アナログメータの回転軸と上限、下限の位置、点灯を検出する輝度差などのパラメータを設定するためのWeb UIを、これまでと同様、PHPで構築した(図6、図7)。



図6 輝度差読取り位置指定 Web UI

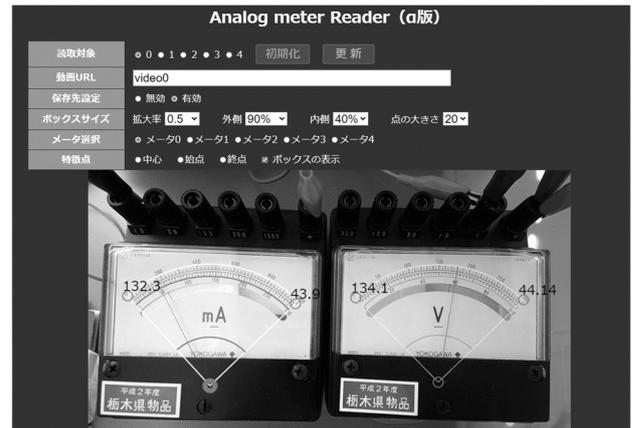


図7 ハフ変換読取り位置指定 Web UI

Web UIで設定された内容を基に、読取りを行うプログラム本体は、Python+OpenCVで構築した。一定時間毎に画像を取得し、読取り結果のテキストをYOLOやVGG19の転移学習にそのまま用いることのできる形で保存する。

次に、転移学習によりモデルを構築するためのプログラムをpythonで作成した。YOLOやVGG19の転移学習は、全ての層の学習を行う事に比べれば低く抑えられるが、計算量が多く、GPUを搭載しない通常のPCでは、結果を得るために多くの時間を要する。このため、GPU搭載PCを用意することなくモデルの作成と評価を行うため、Google Colaboratory¹¹⁾を用いて転移学習を行うプログラムを作成した。

なお、7セグメント表示器やアナログメータの画像は、上下左右が反転すると意味が変わるため、画像水増しする際には注意が必要である。

最後に、ブラウザ上から範囲を指定し、カメラ画像から値を読み取るプログラムを作成した(図8)。

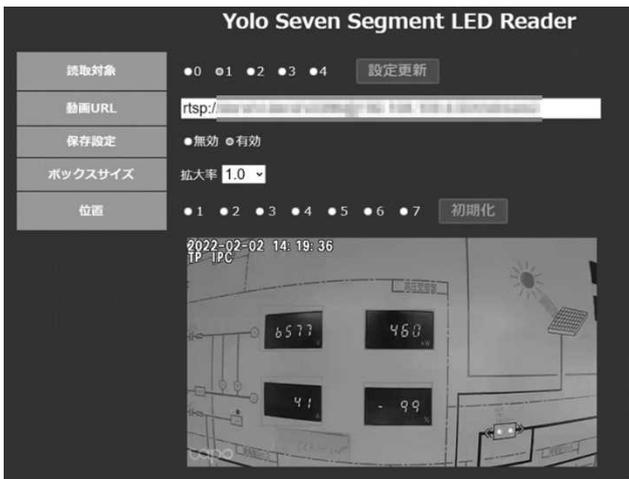


図8 YOLO位置指定 Web UI

3. 2. 2 作成したモデルの実タスクへの適用

今回の一連のモデル作成環境を用いて、実際にセンターの消費電力量計（7セグメント表示器）と、アナログ電圧計、電流計を用いて読取りモデルを作成し、モデルの評価を行った。



図9 昼間と夜間の電力計の見え方

電力量計は、表示が切り替わる瞬間、読み取れない表示（図9左端）や、本来読み取るべきでない表示となることがある。また、昼間と夜間で見え方が大きく異なり、特に夜間は輝度差での読取りに失敗が多くなる。

今回、従来手法で得た教師データを目視で修正し、読取りモデルを作成した結果、高い精度での読取りが期待できるモデルが得られたことから、実際の現場での利用を想定し、NVIDIA社製のGPU搭載開発ボード Jetson Nano 上で画像の YOLOv5n による読取り実行した。読取り結果は MQTT Publisher として、データ収集システムの MQTT Broker に送信する。データ収集システム内では Zabbix Agent2 が MQTT Subscriber として、読取り結果を MQTT Broker から受け取り、Zabbix server がそれを監視しデータを格納する。その後、格納されたデータを、Grafana を通じて可視化する一連の動作を確認した。結果を図10に示す。

図10上の従来手法では、夜間の読取りが上手くいかず、途切れたり、同じ値を繰り返したりしているのに対し、図10下の YOLOv5n モデルを用いた手法では、昼夜問わず安定して読み取れている様子がうかがえる。アナログメータについては、メータの最小値を0、最大値を1として、ハフ変換を用いた手法により教師データを作成し、データ水増しの上で転移学習を行い、モデルを生成した。

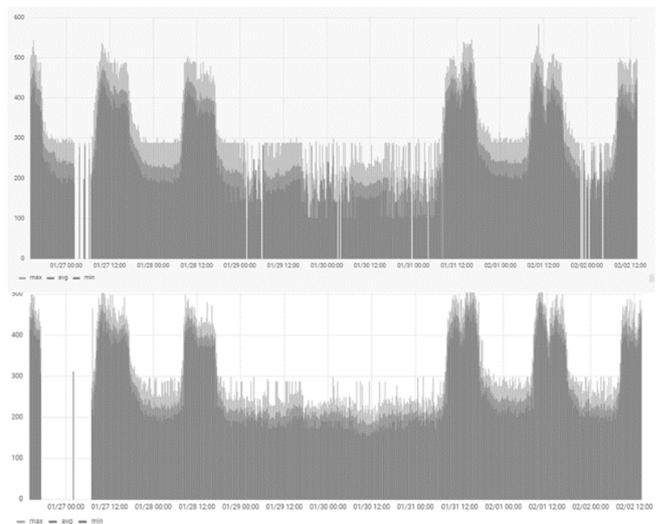


図10 消費電力計の読取り

上が従来手法、下が YOLOv5n モデルを用いた手法ところが、今回生成できた読取りモデルについては、外乱への強さは確認できたものの、目視による読み値とモデル推論値の間に、最大で 0.1 程度の開きがあり、精度に課題が残った。

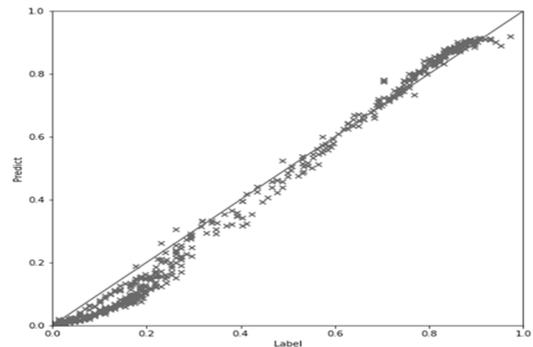


図11 アナログメータ読取りモデルの評価

4 おわりに

今回の研究で以下のことに取組み、これまでのセンサを用いて収集するものに加え、計器類の画像からデータも収集し、ユーザが表示したいデータを、希望する形式で表示できる環境を構築した。

- (1) オリジナル見える化ツールのダッシュボード表示変更に必要なテキストの編集をブラウザ上から行えるようにすると共に、Grafana, Zabbix等の外部ツールを用いた見える化に取り組み、プログラムレスで必要に応じたデータの表示ができる環境を構築した。
- (2) ブラウザ上からの設定で、従来手法を用いてアプリケーションデータを作成するツール、得られたデータから Google Colaboratory 上で推論モデルを生成するツール、作成したモデルを用いて、画像から値を読み取り、csv形式のファイルやDBにデータ保存するツールを作成した。

(3) これらのツールを用いて得られたアノテーションデータから、実際にセンターの消費電力量計やアナログメータの読み取りを行うモデルを構築、継続的に値を収集し、その有効性を確認した。

参考文献

- 1) “所内設備のIoT化に関する研究” 栃木県産業技術センター研究収録, pp. 58-61, 2018.
- 2) “AI導入に向けたデータ収集システムプロトタイプの開発” 栃木県産業技術センター研究収録, pp. 27-30, 2019.
- 3) “ものづくり現場におけるAI活用に向けた分析・解析ツールの開発” 栃木県産業技術センター研究収録, pp. 35-41, 2020.
- 4) Google “Google Data Portal”
<https://marketingplatform.google.com/intl/ja/about/data-studio/>
- 5) Microsoft “Microsoft Power BI Desktop”
<https://powerbi.microsoft.com/ja-jp/>
- 6) Grafana Labs “Grafana”
<https://grafana.com/>
- 7) Zabbix社 “Zabbix”
<https://www.zabbix.com/jp>
- 8) ultralytics “Yolo v5”
<https://github.com/ultralytics/yolov5>
- 9) “Keras Documentation”
<https://keras.io/ja/applications>
- 10) “labelimg”
<https://github.com/tzutalin/labelImg>
- 11) Google “Google Colaboratory”
<https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja>