# IoTを活用した所内装置監視システムの構築

中　川　高　晃*1
小　山　洋　太*2

［要　　旨］

ものづくりの現場におけるIoT導入の障壁となっている事項として、手軽にIoTのシステムを検証できないことなどが挙げられる。今回、安価で簡便なIoTの活用事例検討として、所内にある信号出力がない冷凍庫の温度表示（7セグメントディスプレイ）をデジタル値として取得することを試みた。その結果、実際に現場にいなくても、装置のモニタリングが可能となった。

## 1　はじめに

現実世界の様々なものがインターネットにつながるIoTは多様な場面で活用されており、ものづくりの現場においても生産効率の向上などを目的としたIoTの導入が盛んになっている。IoTを用いることで、データの取得による作業の見える化や得られたデータの分析が容易になり、これまで漠然としていた作業の最適化や新たな課題の気づきにつながる。一方で、どのような課題に対してIoTを導入すべきか、またその費用対効果への懸念といったハードルなどがあり、本格的な導入の前に、手軽に検証できる環境が重要である。

そこで今回、安価で簡便なIoTの活用事例検討として、メーカーによるサポートが切れた信号出力のない所内の冷凍庫の温度表示（7セグメントディスプレイ）を対象に、現場にいなくてもデータのモニタリングが可能なシステムを検討した。

## 2　システム構築について

### 2．1　システム概要

対象となる冷凍庫の温度表示の前に設置したモニタリング状況を図1に示す。シングルボードコンピュータであるRaspberryPiとカメラを温度表示ディスプレイの前に設置して、定点観察を行った。RaspberryPiは所内LANに接続されており、離れた部屋からも遠隔操作可能である。温度は7セグメントディスプレイで表示されており、これをデジタル値としてデータを読み取る手法として、①特徴量マッチング、②OCR、③セグメント有無判定の3つの手法を検証した。

プログラミング言語にはpythonを利用し、カメラで取得した画像の処理にはOpenCV[1)]を利用した。OpenCVは画像映像処理ライブラリで、2値化やフィルタ処理の他、物体や顔の認識など多様なアルゴリズムが用意されており、無償で利用が可能である。ここではバージョン4.5.4.60を用いて検証を行った。

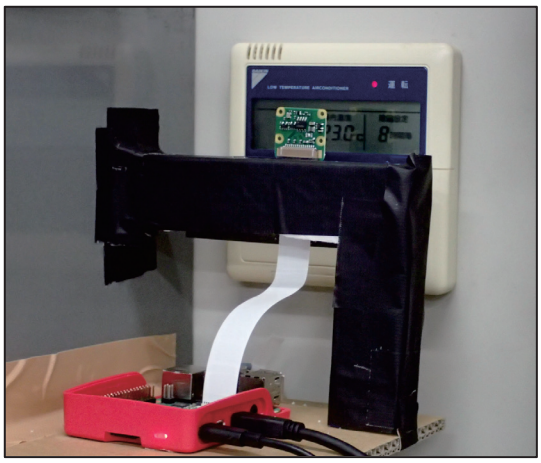

**図1　モニタリング状況**

＊1　応用技術課　技師
＊2　応用技術課　副主査

## ２．２　データの読み取り

### ２．２．１　特徴量マッチング

特徴量マッチングとは、物体の輪郭といった特徴的な箇所に着目し、比較したい画像との特徴一致度合いから、画像の類似度を判定するものである。ある程度のスケール変化や回転変化があっても、画像の比較が可能である。

今回の検証では、モニタリング時の撮像画像と、予め温度毎に用意した全ての画像とで類似度を比較して、最も類似度が高かった結果から温度を判定した。検証結果を図２、３に示す。なお、線でつながれた点と点が、特徴が一致したと判定された箇所であり、特徴の一致度合いが高い順に抜粋して表示している。

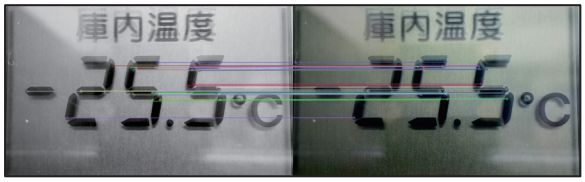


**図２　特徴量マッチング結果（１）**
**（左：撮像画像　右：最も類似度が高い画像）**

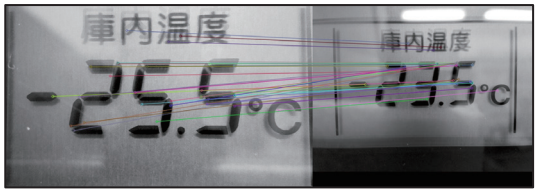


**図３　特徴量マッチング結果（２）**
**（左：撮像画像　右：最も類似度が高い画像）**

図２は、比較対象となる温度毎の画像を、モニタリングカメラ位置と同じ位置から撮像収集した場合の結果である。モニタリング時の撮像画像-25.5℃と比較して、最も類似度が高いと判定された画像も-25.5℃となり値が一致した。一方で、特徴量マッチングの利点であるスケール変化耐性に注目して、比較対象となる画像を変えたところ、一部で誤判定が見られた。その例として、図３は比較対象となる温度毎の画像を、カメラ位置をずらして撮像収集した場合の結果である。モニタリング時の撮像画像-25.5℃に対して、類似度が最も高いと判定された画像は-23.5℃となり、誤った判定を返した。誤判定の要因として、スケール変化のある比較画像はスケール変化のない比較画像に対して撮像画像との類似度が低下している事に加えて、7 セグメント表示はセグメント間に空白があり、全体を１つの数字としてではなくセグメント毎に特徴の比較判定がされた影響によるものと思われる。

### ２．２．２　OCR

OCR は光学文字認識とも呼ばれ、画像から文字や数字を読み取って電子テキスト化する機能である。今回、tesseract[2)] と呼ばれるライブラリのバージョン4.1.1を利用して、モニタリング時の撮像画像から数字の読み取りを検証した。tesseract には学習機能があり、正解と判断する学習用データを用意することで入力画像の認識率が向上するが、学習に有益なデータを用意するハードルから、今回は予め用意されている標準ライブラリを用いて検証を行った。図４に撮像画像と OpenCV により２値化およびノイズ処理を行った画像を示す。


モニター撮像　　画像処理

**図４　OCR 入力画像**

これを tesseract に入力した結果、「-23.5」が出力され値が一致した。一方で、サンプル数を増

やして精度の検討を行った結果、誤判定となるパターンもあった。図５は、tesseract に入力した画像のうち、上段が値が正しく出力された画像、下段が誤った値が出力された画像である。

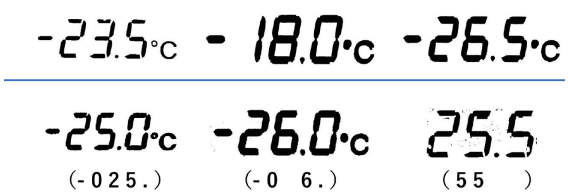


**図５　検証画像（上段：正判定となった画像**
**下段：誤判定となった画像。（ ）は出力値 ）**

誤判定の要因として、7 セグメント表示という角張った数字形状であることと、セグメント間の空白に起因するものではないかと思われる。セグメント間の空白を埋めるように、OpenCV を利用して画像処理による補正を試みたが、元から画像に映っている輝度の濃淡によって、一方のパラメータを補正すると他の箇所でノイズが目立つ結果となり、最適なパラメータは数字毎に異なる結果となった。ただし、先に述べたとおり tesseract には学習機能があるため、学習用データを用意すれば、形状やセグメント間の空白に影響されず、判定精度は向上するものと思われる。

### ２．２．３　セグメント有無判定

表示される数字毎に点灯するセグメントが決まっているという前提から、図６に示すように、７つの領域に分割のうえ各セグメントに順番をつけ、点灯の有無（有は「1」、無は「0」として処理）を判定することによる数字の読み取りを検証した。例えば、図６の場合、数字の８はセグメント番号１から７の順番で表記すると、[1, 1, 1, 1, 1, 1, 1]のように表現され、数字の 1 が表示されている場合は、[0, 1, 0, 1, 0, 0, 0]のように表現される。なお、領域の分割方法は、他のセグメントが干渉しないようにするため、セグメントを横断する形で設定した。

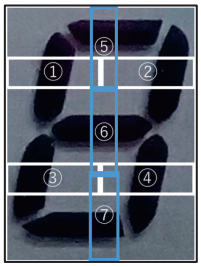


**図６　セグメント有無判定方法**


**図７　セグメント有無判定の流れ**

図７に示すとおり、カメラで撮像したモニタリング画像を OpenCV により輝度補正などの画像処理をした後で、各数字に切り分けてセグメントの判定処理を行った。セグメントの有無判定は各領域における黒ピクセルの量で判定しており、閾値を設定することで多少のノイズに対しても耐性が見られた。判定結果は時刻とセットで csv で保存しており、図８にモニタリング結果を示す。冷凍庫は８時間に１度霜取運転が動作しているが、問題なくモニタリングすることができた。得られたデータから、冷凍庫の温度調整が-23.0℃から-26.5℃において制御されていることがわかった。

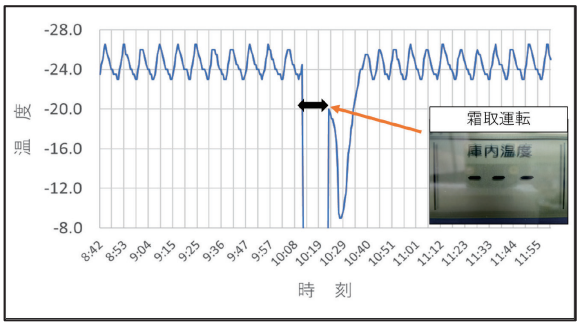


**図８　モニタリング結果**

モニタリング結果はcsvで保存している他に、カメラで撮像した画像も軽量化のうえ保存している。取得データに異常値があった場合にはその時の状況を後からも確認できるようになっている。霜取動作後にモニタリング温度が一度上昇してから下降している時間について、その時の撮像画像を確認したところ数値読み取り結果と一致しており、仕様による挙動と思われる。モニタリング装置は所内LANに接続しているため、図９のように得られたデータを離れた部屋からリアルタイムで確認することも可能である。

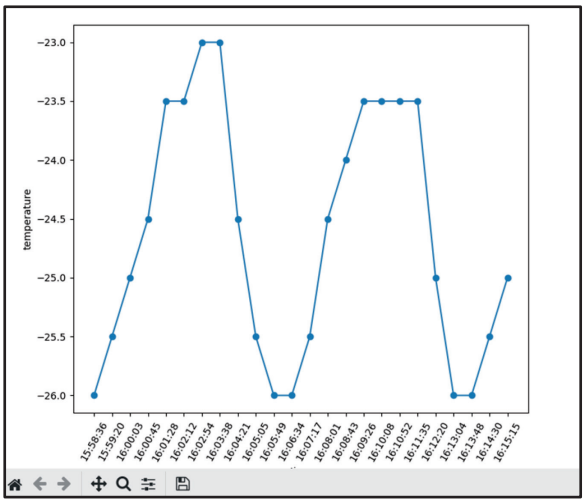


**図９　リアルタイムモニタリングの様子**

## ３　まとめ

所内の冷凍庫の温度表示（７セグメントディスプレイ）を対象に、手軽にIoT導入の検証ができるような、安価で簡便な構成によるモニタリングを試みた。温度表示画像からのデジタルデータの取得方法として、①特徴量マッチング、②OCR、③セグメント有無判定の３つの手法で検証を行った結果、③により現場にいなくてもデータのモニタリングが可能となった。IoT の活用は対象とする物や、周囲の環境・制限などによって、目的達成のための手段は様々である。近年、ハードウェアでもソフトウェアにおいても利用できる高機能なリソースは増えており、一度に大きなコストを掛けて高機能なシステムを構築せずとも、まずは身近な所からIoT導入に関する検証を行うことで、新たな気づきや要望が明確化することができ、本格的な導入に向けた検討が可能となる。

**（参考文献）**

１）https://opencv.org/
２）https://github.com/tesseract-ocr
３）永田雅人，豊沢聡：実践 OpenCV4 for Python 画像映像情報処理と機械学習，株式会社カットシステム