

# 製造現場における装置監視用 IoT システムの開発

中村卓・布施嘉裕・小谷信司

## Development of IoT System for Device Observation of Manufacturing Machines

Takashi NAKAMURA, Yoshihiro FUSE and Shinji KOTANI

### 要 約

全ての機器がインターネットに接続し (IoT: Internet of Things), ビッグデータを駆使しながら機械と人とが連携して実現する製造現場の最適化 (第 4 次産業革命) が注目されている。県内においても設備の IoT 化や IoT 機器の新製品開発に興味のある企業が出始めている。しかし, 中小企業では IoT 機器導入の効果がつかめず, また IoT のために加工等の性能に問題のない高額な工作機械や装置を更新することが難しいため, IoT への設備投資も躊躇している現状がある。本研究では, 安価な組み込みコンピュータやカメラ, センサを用いるとともに, 画像処理, ディープラーニング等のオープンソースソフトウェアを活用し, 県内中小企業が試験的に導入しやすい装置監視用 IoT システムを試作し, 適用例を示すことで, 実現可能性や課題の検討を行った。

### 1. 緒 言

IoT という単語は 1999 年頃から使われ始めた<sup>1)</sup>が, IoT のコンセプトはそれ以前から提唱されている<sup>2)</sup>。近年では, PC の高性能化や小型化, カメラやセンサ類が安価になったことで IoT 技術の発展が著しい。

製造現場においても, IoT 技術の導入が大手企業や大規模工場などを中心に広がっており, 多くの県内中小企業も興味を持ち始めている。県内企業では, 製造現場において, 従来は目視で行ってきた装置の監視作業を IoT によって自動化するとともに, 生産過程の見える化を行い, 生産効率を上げたいという要望が多い。

IoT に興味を持つ企業が増えてきた一方で, 導入には躊躇している現状がある。その原因として, IoT 導入による具体的な効果が見えないことや導入コストの問題などがある。例えば, 製造装置の自動監視や生産過程の見える化を行う場合, 装置をネットワークに接続してサーバなどにデータを蓄積する作業などが必要になるが, 装置自体が古い場合ネットワークに接続できないケースが多く, ネットワーク接続するためには装置の更新が必要となる。この場合, まだ正常に稼働している装置を更新することになるため, 装置の更新を躊躇し, IoT 技術の導入を見送ることが多い。その結果, IoT に関する企業の関心やニーズは非常に高いが, 現場レベルでの導入が進んでいないという現状がある。

カメラやセンサなどから得られたデータの解析にも課題がある。製造装置の監視において, カメラなどから得られた画像に対して処理を施して必要な情報を収集する

必要があるが, 一般的な画像処理による手法では, 環境変化に応じてパラメータを変更する必要があり, パラメータの調整などに専門的な知識が要求される。その一方で, 画像認識に近年盛んに研究が行われているディープラーニングを用いた場合, 学習データ中に環境光の変化など環境が変化した画像を混ぜて学習することで, 環境変化に対して頑強にすることができる。また, 学習する際に環境の変化なども考慮するため, 細かなパラメータ調整もあまり必要としない。そのため, 製造現場において画像認識のためにディープラーニングを用いる利点は大きいと考えられる。

本研究では, これらの問題を解決するために, 県内企業が試験的に導入しやすいディープラーニングを用いた装置監視用 IoT システムの試作を行い, 実際に稼働している装置への適用を行った。さらに, システムの試作や現場への適用を通じて, 製造現場などに IoT システム導入時の実現可能性や, 課題などの考察を行った。

### 2. 装置監視用 IoT システム

#### 2-1 システム概要

我々が試作した装置監視用 IoT システムの概要を図 1 に示す。本システムは, 「監視対象のモニタリング部」と「ディープラーニングによる解析部」から構成される。「監視対象のモニタリング部」では, カメラやセンサなどを用いて監視対象のデータの取得を行い, 取得データの蓄積を行う。「ディープラーニングによる解析部」では, 得られたデータに対してディープラーニングを用い

て解析を行い、その結果を出力する。これらはネットワークストレージによって接続されており、カメラやセンサで取得したデータの送受信はこのネットワークストレージを介して行われる。

カメラやセンサ、組み込み PC などは Raspberry Pi など市販されている安価なもので構成し、処理に必要なライブラリやネットワークストレージなどはフリーで利用可能なものを利用した。ディープラーニング用の PC についても、一般に市販されている PC のスペックで十分に処理が可能である。安価なデバイスやフリーのライブラリなどを利用することで導入コストの削減を可能として、中小企業が試験的に導入しやすいシステムを実現した。

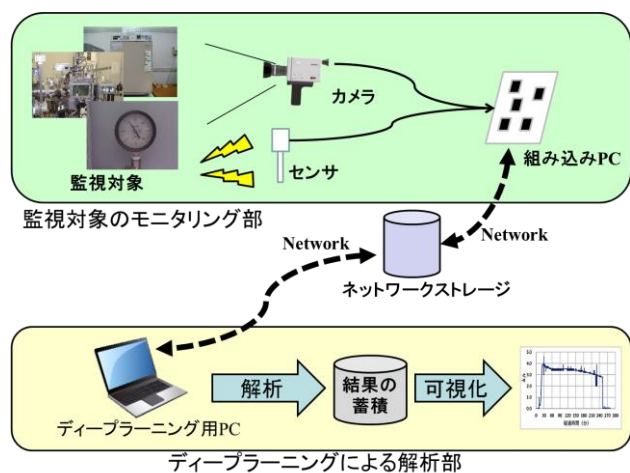


図1 装置監視用 IoT システム概要

## 2-2 監視対象のモニタリング部

「監視対象のモニタリング部」では、監視対象の装置などの様子を装置の外部からカメラやセンサを用いて取得するため、カメラやセンサ、及びそれらからデータを取得しネットワークストレージに送信する組み込み PC から構成される。図2は、当センターが保有するチラーユニットの水圧計をモニタリングした例である。保有するチラーユニットの調子が悪いという相談を所内で受けた。その原因調査として、チラーユニットの水圧の変化を記録するために水圧計の撮影を行った。図2の吹出し内の写真は実際に撮影された画像である。図2では、USBカメラで水圧計を撮影し、撮影画像を Google Drive へ送信している。これらの制御は全て Raspberry Pi で行っている。Raspberry Pi では、一定時間ごとに撮影、Google Drive へアップロードを繰り返すスクリプトを作成、実行することでモニタリングを行った。本例では、夜間でも撮影を行う必要があるため、赤外線 LED と赤外線カメラを用いて撮影を行った。

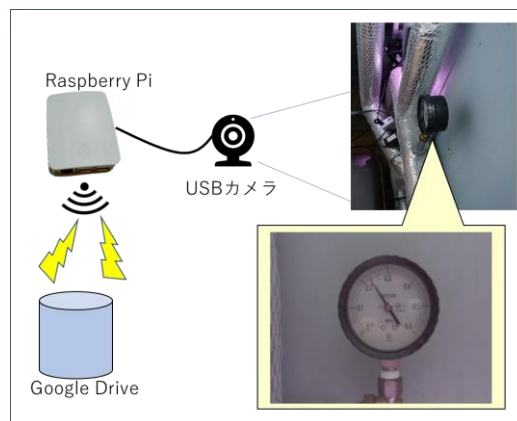


図2 監視対象のモニタリング部

(チラーユニットの水圧計を撮影している様子)

## 2-3 ディープラーニングによる解析部

モニタリング部で得られたデータは、逐次ネットワークストレージに送信、蓄積される。得られたデータはネットワークストレージ上に存在するため、直接データにアクセスして確認することも可能である。しかし、データは時間の経過と共に膨大になっていくため、得られた膨大なデータに対して処理を行い、必要な情報のみを効率よく収集する必要がある。これらのデータを効率よく処理するため、ディープラーニングを用いた解析をモニタリングとは別に行う必要がある。

ディープラーニングを用いた解析では、事前に学習データから学習モデルを作成する必要がある。「ディープラーニングによる解析部」は、図3に示す学習モデルを作成するための「学習モデル作成部」と、実際に解析を行う「データ解析部」から構成される。なお、これらは Python とディープラーニングのフリーのライブラリである Chainer を用いて実装を行った。

「学習モデル作成部」では、まずネットワークストレージ上に蓄積されたデータから学習モデルを作成するための学習データを作成する。必要な学習データは解析したい内容によって異なる。例えば、モニタの数字を認識したい場合には0から9までの数字画像を集めた学習データが必要になる。学習モデルの作成手順は、まず蓄積したデータ中の必要な箇所を指定し、その部分の情報を PC で処理し、学習データとして収集する。十分な数の学習データを収集した後、それらを用いてディープラーニングで学習し、学習モデルを作成する。

「データ解析部」では、学習モデル作成に利用したデータとは異なるデータに対して、作成した学習モデルを適用し、その結果を出力する。出力結果については、CSVファイルなどの形式で記録し、蓄積する。蓄積された出力結果を基に、利用者は異常検知や、結果の可視化などを行う。例えば、異常値が得られた場合には、現場に

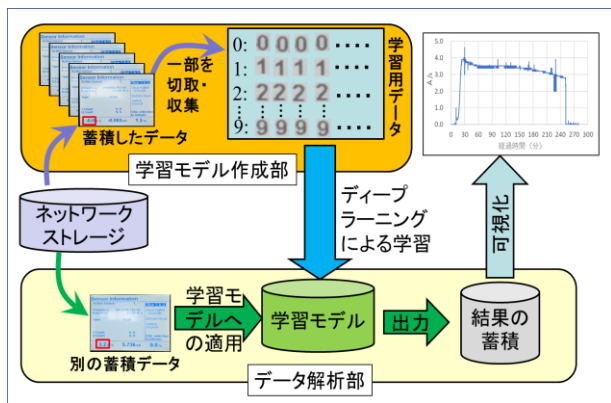


図3 「ディープラーニングによる解析部」の構成

アラートを発生させたり、得られたCSVファイルなどを用いて結果をグラフ化したりすることができる。

### 3. 製造現場への適用例

#### 3-1 製造装置のセンサモニタへの適用

試作した装置監視用 IoT システムについて、県内の企業で実際に稼働している製造装置のセンサモニタへの適用を行った。図4はその時に撮影した画像とその画像を基にディープラーニングを用いて必要な箇所を抽出し、グラフ化したものである。

撮影は、センサモニタの正面に USB カメラをモニタ画面全体が写るように設置した。センサモニタの様子は動画ファイルで取得し、解像度は 1280×720 ピクセルで 30 fps、総フレーム数は 506,124 フレーム (約 4 時間 40 分) のデータを利用した。図4の左の画像は実際の撮影画像である。

図4の画像中の左下の口の箇所を切り出して、各位 (一の位と小数点第一位) の数字をディープラーニングで認識し、各位の認識結果を合成して数値として出力した。なお、数字1つあたりの画像サイズは 40×50 ピクセルであった。図4の右のグラフはその結果をグラフ化したものである。グラフの横軸は時間、縦軸は認識したセンサモニタの値となっている。ディープラーニングの処理については、OS:Windows10, CPU: Core i7-6700(3.4 GHz), RAM:32 GB,HDD:2 TB の PC を用いてオフラインで処理 (撮影を行った後に、その動画ファイルを用いて解析) を行った。処理速度については、1秒間に約 210 フレームであり、図4で利用した動画ファイルの全フレームを処理するのに約 40 分要した。

図4のケースでは、学習モデルを作成するために、学習データとして各数字 (0 から 9) の画像をそれぞれ 10,000 個用意した。ディープラーニングにおいて必要な学習データ数は十分な精度を出すために、ひとつの事象につき 5,000 個以上必要だといわれている。そのため、

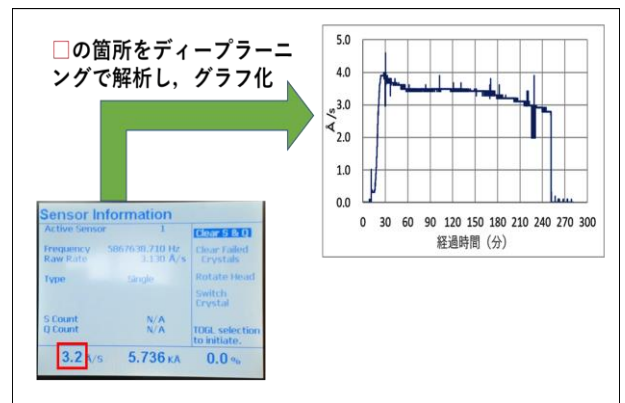


図4 センサモニタへの適用結果

本ケースではより高い認識精度を得るためにその倍の 10,000 個とした。なお、学習で利用した画像は全て図4で利用した動画ファイルとは別の日に撮影したものを利用した。

#### 3-2 検出精度の検証

ディープラーニングによって得られた結果の認識精度についても検証を行った。真値は、図4で数字の認識を行った口の箇所に対して各位 (一の位と小数点第一位) の数字を画像処理 (事前に作成した 0 から 9 の数字の画像とのテンプレートマッチング) により 0 から 9 の数字を認識し、その結果を合成して出力したものを利用した。検証は、作成した真値とディープラーニングによる解析結果とを比較することで行った。また、認識精度についても算出を行った。

図5は、ディープラーニングによる解析結果と真値とを比較したグラフである。図4と同様、横軸が経過時間、縦軸が各認識方法で検出したセンサモニタの数値である。認識精度については、ディープラーニングによる手法は、本ケースでは 0.99 以上の精度があることが確認された。なお、認識精度については、 $1 - (\text{真値と異なっていた数値のフレーム数} / (\text{使用した動画ファイルの全フレーム数} (=506,124 \text{ フレーム}))$  で算出した。

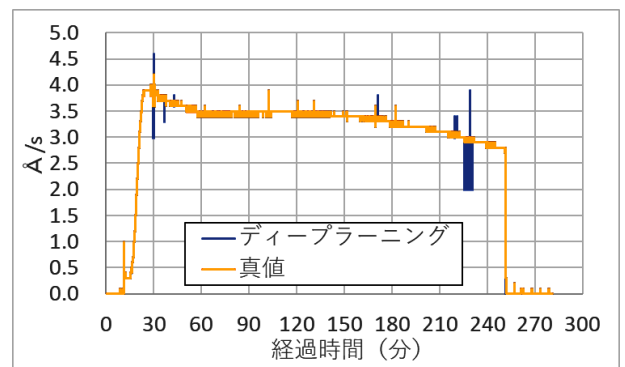


図5 真値との比較結果

### 3-3 学習データ数の違いによる検出精度の検証

3-1より、本ケースでは、事前に十分な学習用データが用意できれば、十分な精度が得られることが確認された。しかし、学習データとして1数字あたり10,000個の画像を用意することは負担が大きいため、収集する学習データ数を削減する必要がある。学習データ数が減少させた場合、認識精度の減少が懸念される。そこで、学習データ数を減少させた場合、認識精度にどのぐらいの差が現れるかについても検証した。検証方法は3-1と同様に、ディープラーニングによる認識結果と画像処理（テンプレートマッチング）による認識結果を比較することで行った。学習データ数については、100・1,000・5,000・10,000個（1数字あたり）の4パターンで行った。

図6は、学習データ数を変化させて3-1と同様の実験を行った結果である。横軸は使用した学習データ数、縦軸は認識精度である。図6より、学習データ数が5,000個以上の場合には非常に高い認識精度であるが、学習データ数が少なくなるほど認識精度は著しく減少した。特に学習データ数が100個の場合には、認識精度は2割程度まで減少した。

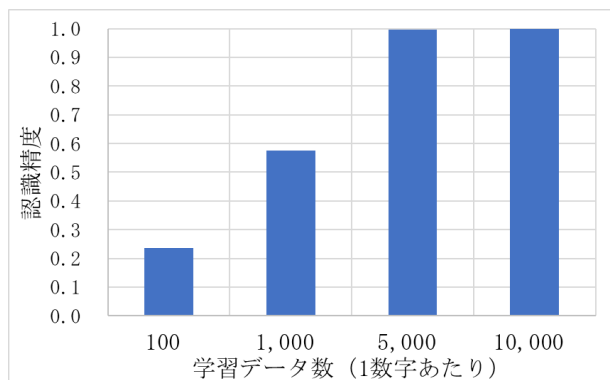


図6 学習データ数による認識精度の違い

## 4. 考察

3章の結果より、試作したシステムを用いることで、装置外部からデータを取り出すことが可能であることが確認できた。また、本研究で試作した装置監視用IoTシステムは安価な組み込みPCやセンサ類などを利用することで導入コストを低くし、製造装置の外から監視を行うことができるため、設置コストも抑えることができた。これらのことから、本試作システムを用いることで、中小企業が試験的に導入などの目的で利用可能であることが分かった。

ディープラーニングを利用する際の課題として、学習モデルを作成する際に用意する学習データ数と認識精度、及び学習データを収集する際の負担の増大の問題がある。3章で適用したセンサモニタの場合、0から9までの数字を学習する必要がある。十分な認識精度を得るために

は、図6の結果より1数字辺り5,000個必要であることが分かるが、実際に必要数(5,000×10=50,000個)を集めるためには、多大な労力と時間がかかってしまう。そのため、学習データ収集に費やす負担の軽減が課題として挙げられる。データ収集の負担軽減のための方法として、得られたデータに対し、画像処理などを用いて学習用画像を増やす方法(データ拡張)などが考えられる。例えば、得られた画像に対して、拡大・縮小、回転、平行移動などを用いることで、ひとつの画像から複数の画像を作成することができるため、データ収集の負担軽減が期待される。データ拡張手法については、監視対象や抽出するデータの内容によって効果的な手法が異なることが考えられるため、そのようなケースでどの手法が有効であるかなどについて、今後、検討を行っていく必要がある。

## 5. 結言

本研究では、県内中小企業が試験的に導入しやすい装置監視用IoTシステムの試作を行い、実際の製造装置に適用した。その結果、試作したシステムにより、安価で設置も容易でありながら、装置外部から情報を取得することが可能であることが確認できた。その一方で、ディープラーニングを利用するために必要な数の学習データの取得方法に工夫が必要であるなどの課題が得られた。

今後の展望としては、現段階では適用例が少ないため、当センターが実施している各種事業を通じて様々な製造現場で本試作システムを活用し、適用例を増やしていくことが挙げられる。また、学習データ収集の負担軽減のためのデータ拡張手法についても検討を行っていく。

## 参考文献

- 1) Kevin Ashton: "That 'internet of things' thing in the real world, things matter more than ideas", RFID Journal, June (2009)
- 2) Mark. Weiser: "The Computer for the Twenty-First Century", Scientific American, pp.94-104 (1991)