エッジAI向け異常検知モデルに関する研究

本間　稔規、全　　慶樹

Research on Anomaly Detection Models for Edge AI

Toshinori HONMA, Keiki ZEN

抄　録

近年、生産者人口の減少から人手不足が深刻であり、製造業において省人化、自動化のニーズが高まっている。一次産業が基幹産業である北海道では、農水産物の原料受入検査や食品製造業における異物の目視検査など、人手のかかる作業の自動化が望まれている。このため、AIを組み込んだ検査機械の開発などが活発に行われている。これまではエッジデバイス（たとえば、Raspberry Piなどの小型の組込計算機）で収集したデータをサーバー側のAIで処理していたが、通信による遅延のためリアルタイム処理に向かないこと、エッジデバイスの数が増えるとネットワークの通信帯域がひっ迫すること、インターネット経由でデータをクラウドに送信することがセキュリティ上のリスクになるなどの問題があった。最近ではエッジデバイスにAIを実装し装置に組み込むケースが増えてきているが、計算能力の関係から学習はクラウド上のサーバーなどで行い、構築した学習モデルをエッジデバイスに組み込んで推論用として用いている。このようなAIを現場で運用する場合に、最初に組み込んだ学習モデルのままでは取り扱うデータの傾向の変化やセンサの経年変化などに対応できずに生じてしまう予測性能の劣化が問題となっている。

本研究では、予測性能が劣化する前に、エッジデバイス上で学習モデルを適宜更新する機能を実現することを目指して、少数の訓練データのみの学習で学習モデルを構築する手法である、スパースモデリングやリザバーコンピューティングを用いた異常検知技術の開発を行った。画像や分光データ、音声などの少量のデータを対象として異常検知モデルを構築し、評価を行ったところ、少量の訓練データで十分な性能が得られることがわかった。

キーワード：異常検知、エッジデバイス、スパースモデリング、リザバーコンピューティング

Abstract

A shortage of workforce due to decreasing working age population have been getting severe in recent years, therefore, a lot of demands for labor-saving systems or autonomous systems are growing. Especially in Hokkaido, where major primary industry consisted of agriculture and fisheries, autonomous inspection systems for detecting foreign matter or deteriorations among agricultural crops or fishes in sorting facilities or food factories are strongly desired. Inspection machines have been evolved to have high performance in inspecting products, because of cutting edge deep learning implemented in those devices. Early AI systems implemented in edge devices require communication with cloud server to get prediction results, this procedure results in decreasing real-time performance because of network latency, also in short bandwidth on a network because a substantial number of edge devices all at once connect to network. In these days, AI estimation models build on cloud server have been implemented in edge devices such as Raspberry Pi. The performance of prediction with first implemented AI model is tend to gradually decline along with time passing, because of data drift or sensor deterioration over time. To maintain performance, updating AI model only in the edge device is getting most important to prevent network congestion.

In this research, we utilize anomaly detection techniques with sparse modeling for image data, and also with reservoir computing for time series data, and examined the performance in the condition of a small numbers of training data. We developed and tested several anomaly detection models, and ultimately achieved satisfactory performance.

KEY-WORDS : Anomaly detection, Edge device, Sparse modeling, Reservoir computing

---

事業名：経常研究

課題名：少量の訓練データから異常検知モデルを構築可能なエッジAIに関する研究

1. はじめに

近年、生産者人口の減少から人手不足が深刻であり、製造業において省人化、自動化のニーズが高まっている。一次産業が基幹産業である北海道では農水産物の原料受入検査や食品製造業における異物の目視検査など、人手がかかる作業の自動化が望まれている。一方、深層学習などによりAIの性能が飛躍的に高まったことを受けて、AIを組み込んだIoT機器の開発が活発に行われている（図１）。最近ではエッジデバイス（たとえば、Raspberry Piなどの小型の組込計算機）にAIを実装し使われるケースが増えてきているが、計算機の能力が限られているため膨大な計算量を必要とする学習用途ではなく、主として推論用として用いられている。このようなAIを現場で運用する場合、エッジデバイス上では学習モデルを更新することが困難なことから、取り扱うデータの傾向の変化やセンサの経年変化などが原因で生じる予測性能の劣化が問題となる。これを防ぐためには、予測性能が劣化する前にエッジデバイス上で学習を行い学習モデルを適宜更新することが重要である。

本研究では、エッジデバイス上で学習・推論の機能を実現することを目指して、少数の訓練データのみの学習で十分な予測性能を実現可能なスパースモデリングやリザバーコンピューティングによる異常検知技術に関して開発を行った。

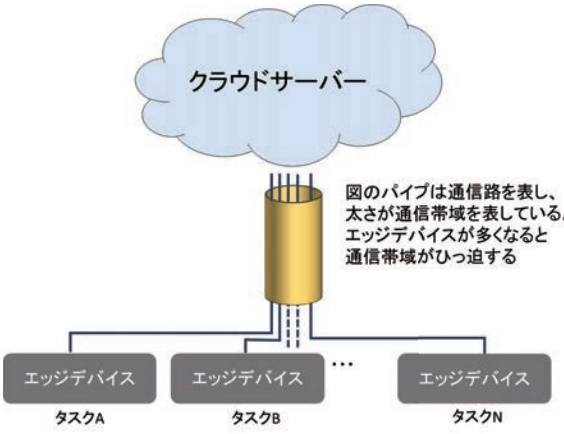

図１　エッジデバイスとクラウドサーバーのネットワークのイメージ

2. エッジデバイスへのAIの実装

2.1　エッジデバイスでのAI機能の実現方法

エッジデバイスにおいてディープラーニング等のAI機能を実現する方法の一つは、クラウドサーバーでAI機能を実行し、エッジデバイスはセンシングのみを行うというものである（図２）。初期のディープラーニングは学習モデルが大きいため、当時のエッジデバイスに実装できないことが多かった。そこで、エッジデバイスではデータの送受信だけを行い、AI推論はサーバーで実行する形態がよく使われた。この場合は通信時の遅延が発生するため、リアルタイム処理が困難であった。最近はディープラーニングの研究が進み、モデル圧縮など軽量化技術が進展したことやエッジデバイスの性能向上により、エッジデバイスにAIモデルを組み込んで推論機能を実行することが可能となってきた（図３）。すなわち、エッジデバイスでデータをセンシングし、内蔵するAIで推論を行い、その結果にもとづいて制御を行うことができるようになった。この場合、AI処理の実行時にサーバーとの通信は発生しないためリアルタイム処理が可能である。

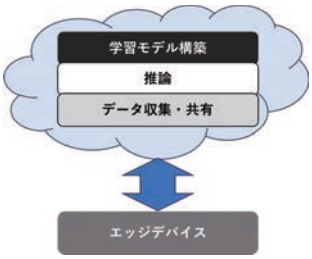

図２　クラウドサーバーで学習モデル構築と推論を実行する場合

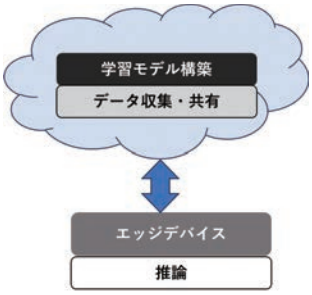

図３　学習モデル構築はクラウドサーバーで行い、推論はエッジデバイスで行う場合

2.2　AIモデルの劣化への対応

ディープラーニングによるAIモデルの学習では、まず訓練データを収集しモデルを構築することから始める。ディープラーニングではニューラルネットワークのパラメータ数が多いため、多くの訓練データが必要となる。すなわち、高い正解率と汎化性を実現するモデルを構築するには大量の訓練データを収集する必要がある。それらを学習して構築したモデルをエッジデバイスに実装して運用することになるが、データの季節変動や年次変動、またはエッジデバイスで使用しているセンサの経年劣化などが原因で機械学習モデルの性能が低下する可能性がある。たとえば、センサの劣化が原因で推論性能が低下した場合はセンサの交換を行うなどの対応も考えられるが、センサの特性にばらつきがある場合は同一型番のセンサでも再度キャリブレーションをし直す必要があったり、さらには機械学習モデル自体を構築し直す必要がある。ディープラーニングの機械学習モデルの構築は計算量

が多いため、GPUなどのアクセラレータを持たない、計算資源の限られたエッジデバイスでは対応が困難である。そのため、ネットワーク経由で訓練データをサーバーに送って機械学習モデルを再構築することになる。しかし、大量の訓練データの送信はネットワークの通信帯域を消費し、さらに、他の多数のエッジデバイスが学習モデルの構築を行うことは、たとえ高性能なサーバーでも負荷の高いタスクである。この問題を解決するには、エッジデバイス上で機械学習モデル構築が実行できるようにすればよい（図４）。具体的にはエッジデバイスの性能を上げる、もしくは計算量のかからないアルゴリズムを使うなどの方法が考えられる。エッジデバイスの性能を向上させる方法として、たとえばGPUを搭載したエッジデバイス（NVIDIA Jetson AGX Orinなど）を使うことが考えられるが、この製品は価格が30万円以上(2025年時点）であり、通常のPCよりも高価になってしまうため、コスト的な合理性は低くなる。

一方、計算量のかからないアルゴリズムとして近年注目されているのがスパースモデリングとリザバーコンピューティングである。スパースモデリングは画像データ等への適用事例が、またリザバーコンピューティングは時系列データへの適用事例が多い。これらのスパースモデリングおよびリザバーコンピューティングによる異常検知技術について次に説明する。

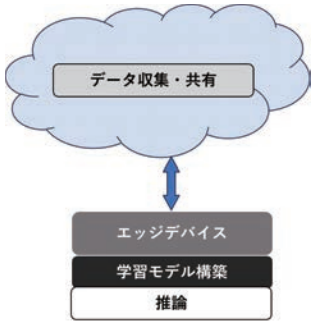

図４　学習モデルの構築、推論をエッジデバイスで行う場合

3. 異常検知手法

3.1　画像データに対する異常検知手法

機械学習を用いて、ある母集団に属するデータの異常を検知するには、その母集団での正常データを学習し、正常データを生成するような学習モデルを構築すればよい（すなわち、入力データと出力データが同じになるように学習モデルをトレーニングする)。この学習モデルに異常のあるデータを入力すると、異常部分は学習していないため、入力データと等しくなるような出力データの生成に失敗する。この生成に失敗した出力データと入力データの差を求めることにより異常を検知する。本研究では、少ない訓練データで異常検知を行うスパースモデリングの手法の有効性を確認するために、オートエンコーダ型のニューラルネットワークモデルであり異常検知が可能なVAE（Variational Auto Encoder）[1]、およびこれをベースとして中間層にスパースコーディング（スパース辞書学習）を組込んだVSC（Variational Sparse Coding)[2]を用い、性能を比較した。さらに、画像分類のニューラルネットワークモデルであるVGG16[3] の中間層の出力を入力データとしてスパース辞書学習を行うMLF-SC（Multi-Layer Features to Sparse Coding）[4] について訓練データ量と推論性能、異常検知性能などの評価を行った。

3.2　時系列データに対する異常検知手法

時系列データ用のネットワークモデルとしてリカレントニューラルネットワーク（Recurrent Neural Network、以下、RNN）がよく用いられる。このニューラルネットワークでは出力層の信号を入力層に接続することにより、時間的に過去の情報を反映したネットワークを構築することができる。リザバーコンピューティングは、このRNNをベースとした機械学習モデルである。リザバーコンピューティングでは中間層がランダムな固定の重みで結合された構造となっている。このようにランダムな結合重みでニューロン同士が結合された中間層はリザバー層と呼ばれ、過去の情報を含んだ高次特徴量を出力する。リザバー層では重みは固定であり学習する必要がなく、出力層のみ学習すればよいため、通常のRNNに比べ格段に計算量が少なくなる。リザバーコンピューティングを用いた時系列データに対する異常検知手法も前述した画像データに対する異常検知手法と考え方は同じであり、目標信号と予測信号の差で評価する。具体的には、入力層に入力したデータについて、一定時間後の出力データを予測するように出力層の重み調整の学習を行う。予測したデータと実際の信号との差が小さければ正常、差が大きければ異常と判断する。本研究ではリザバーコンピューティングの実装としてEcho State Network（以下、ESN）[5] を用いた。

4. VSCによるスパースモデリングの性能評価

VAE（図５）は、事前分布がガウス分布であるデータを対象として、データをその平均と標準偏差およびガウス分布で表現し、内部の潜在変数をランダムサンプリングするオートエンコーダである。VSCは中間層において潜在変数をサンプリングするための事前分布をVAEで用いているガウス分布からSpike & Slab分布（ガウス分布の中央部分の確率を高くした分布）に変更したものである（図６)。この変更によりデータ表現のパラメータを少なくすることが可能である。すなわち、少ないデータで機械学習モデルの構築が可能となる。

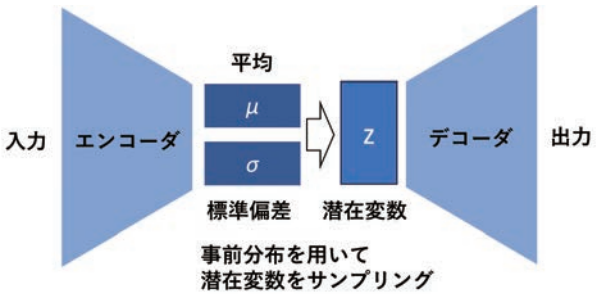

図５　VAEおよびVSCで用いるネットワーク構造

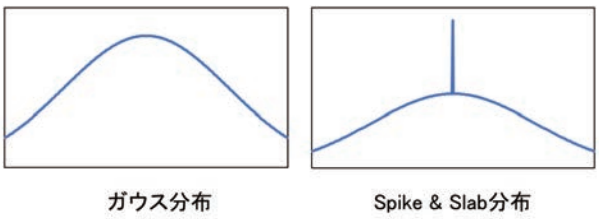

図６　VAEで用いるガウス分布（左）とVSCで用いるSpike & Slab分布（右）

4.1　MNIST手書き数字データセットを用いた性能評価

MNISTは米国のNIST（国立標準技術研究所）が整備している機械学習用のデータセット[6]であり、ディープラーニングのネットワークモデルの性能評価に広く利用される。本研究では手書き数字データセット（訓練データ用：60000枚、テストデータ用：10000枚）の訓練データについて、ランダムに所定の数の画像を抽出し、データセットを作成してVAEとVSCに学習させ、訓練データのサイズとモデルの性能の関係を再構成誤差（ここでは平均二乗誤差、Mean Squared Error：MSE）により評価した。また、同じ条件においてVAEとVSCの学習における計算時間を比較し、リアルタイム処理への影響を評価した。さらに、計算資源が豊富なパーソナルコンピュータ（以下、PC）と計算資源の限られたエッジデバイスでVAE、VSCによる学習を行った場合の計算時間を比較した。

4.2　実験

VAE、VSCの各々のニューラルネットワークのハイパーパラメータ（入力層および出力層サイズ（＝768）、中間層サイズ（＝400）、バッチサイズ（＝10）、エポック数）を同一とし、訓練データ数のみを変更して評価した。どちらもPythonで実装したコードをシングルスレッドで実行した。学習モデルを構築するにあたり、バッチサイズとエポック数を決める必要がある。訓練データ数の最小を50と設定したのでバッチサイズは訓練データ数よりも小さい10とした。エポック数については、事前の予備実験により再構成誤差の減少が収束する最小限の回数とし、ここでは1000に設定して評価を行った（図７、図８）。

訓練データ数を変更しながらVAEとVSCの再構成誤差の変化を調べた（図９）。その結果、訓練データ数が少ない場合はVSCの方が高性能であることがわかった。訓練データ数を50としてVSC学習モデルを評価した場合のMNIST手書き数字の入力と出力を図10に示す。入力画像と出力画像の差が小さく、ほぼ同じ画像が生成できていることから、少ない訓練データで性能のよいAIモデルの構築ができることがわかった。

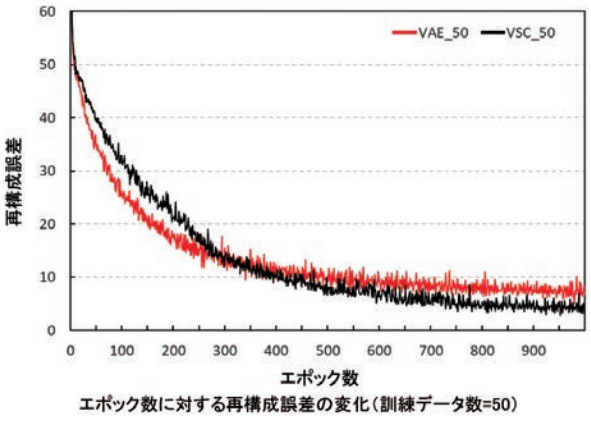

図７　エポック数と再構成誤差の関係（訓練データ数＝50）

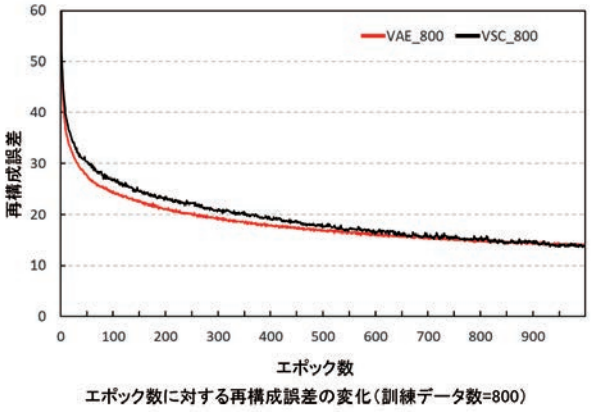

図８　エポック数と再構成誤差の関係（訓練データ数＝800）

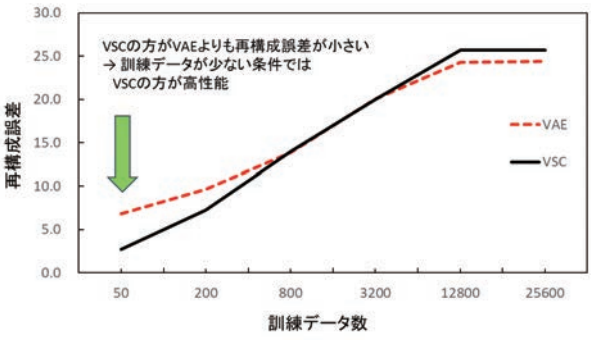

図９　訓練データ数と再構成誤差の関係（VAEとVSCを比較）

図10　手書き数字とVSC（訓練データ数＝50）の学習モデルで推論した手書き数字の比較

4.3　計算時間

VSCとVAEでは、中間層の潜在変数を生成するときに用いる事前分布が異なる。事前分布による計算は学習モデル構築時や推論時で毎回行われるが、VAEで用いているガウス分布よりもVSCで用いているSpike & Slab分布の方が計算ステップが多いため、計算時間がかかると予想される。そこでPC（CPU：Intel Core i7 8700K）とRaspberry Pi 4B+（以下、RPi4）（CPU：ARM Cortex-A72）とを用いて、訓練データ数を変えながらVSCとVAEの計算時間を計測し、それらの比（VSC/VAE）を求めた。その結果、VSC/VAEの値はPC、RPi4のどちらの場合でも訓練データ数によらず1を超えていることから、VSCはVAEよりも計算時間がかかることがわかった（PCで約16%、RPi4で約8%程度）（図11）。VSC自体はVAEよりも10%ほど計算時間が多くなるデメリットがあるが、少ない訓練データで高性能な学習モデルを構築できることから全体的には処理時間の低減を実現できることがわかった。次に、1エポック数当たりの計算時間

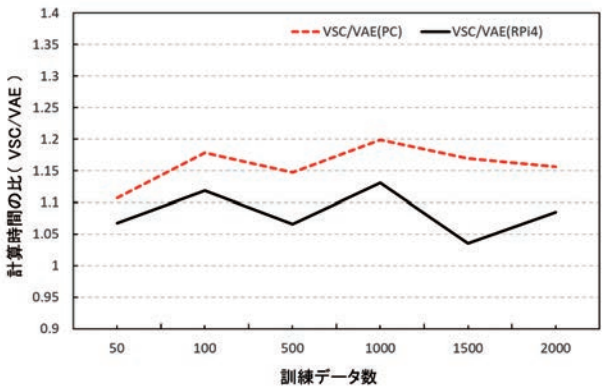

図11　VSCとVAEの計算時間の比較

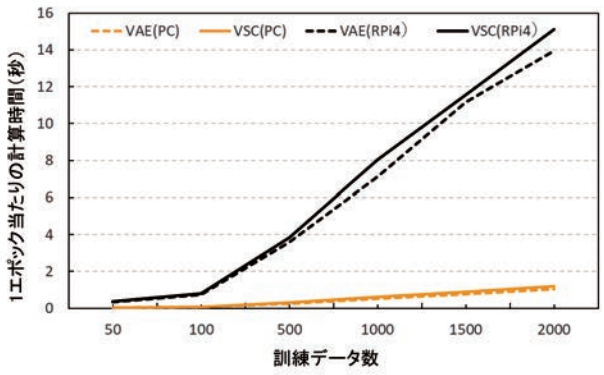

図12　PCとRaspberry Pi 4 B＋の計算時間の比較

をPCとRPi4とで比較した（図12）。訓練データ数を50から2000まで変化させながら計測した結果、訓練データ数によらず、PCはRPi4よりも約13倍高速であることがわかった（計算時間については、たとえば訓練データ数＝50の場合、PCが0.028秒に対し、RPi4は0.372秒）。また、推論時ではPCは1回の推論に要した計算時間は約0.0028秒、RPi4は0.02秒であった。この結果から、RPi4などのエッジデバイスの性能が向上してきているとはいえ、プロセッサやメモリの動作周波数の差から処理能力が大きく異なることがわかった。これらから、エッジデバイスで学習を行う場合は計算量を削減するために、訓練データ数を減らすことは有効であると考える。

5. MLF-SCによる食品混入異物検出

次にスパースモデリングによる異常検知技術により、食品に付着する人毛の検出を試みた。具体的には、スパースモデリングによる異常検知実装の一つであるMLF-SC学習モデルを用い、分光画像データから豚挽肉に混入した異物（人毛）の検出能力について評価した。

5.1　MLF-SCの概要

MLF-SCはディープラーニングの画像分類モデルの一つであるVGG16をベースとし、VGG16の中間層で抽出された複数の特徴量を後段のスパースコーディング部の入力データとして学習を行う（図13）。この中間層に抽出される特徴量は入力画像そのものに含まれる特徴量と比較してノイズ等の不要な情報が減少している。この特徴量をスパースコーディングの辞書データとして学習に用いることで出力層での再構成誤差を最小にすることが期待できる。

MLF-SCを用いた異常検知技術においてもこれまでと同様に、正常なデータのみを訓練データとして使用する。出力画像の再構成誤差が大きくなる部分を異常として検出する。

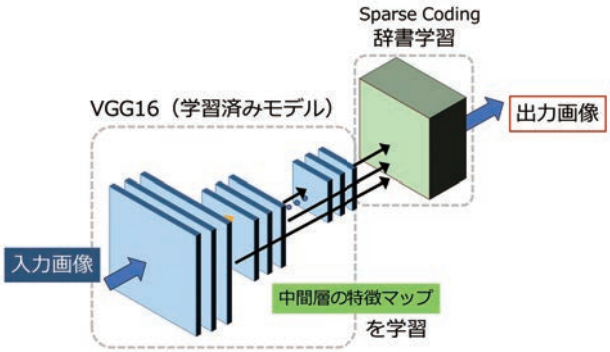

図13　MLF-SCのネットワークモデル

5.2　食品混入異物検出

MLF-SCの評価には、当場が開発した、任意の複数波長の分光画像を撮像可能な多眼式分光イメージングカメラ[7]

（図14）で撮影した、表面に人毛を付着させた豚挽肉のデータを用いた。この多眼式分光イメージングカメラにより、617nm、697nm、796nm、900nmの４波長の分光画像を取得し、画素ごとの分光データを求めた。MLF-SCはVGG16をベースとしていることからRGB画像（３チャネル画像）を入力データとする仕様となっている。そのため、本研究では４波長から３波長を抽出して学習モデルを作成し、最も異物検出性能が良くなる３波長（697nm、796nm、900nm）の組み合わせを選択した。次にこの異物検出用の学習モデル作成の具体的な手順を示す。まず分光画像の豚挽肉部分をブロック領域に分割し、人毛が含まれないブロックを「正常」として学習させて異物検出用学習モデルを作成する。次に人毛が含まれるブロックをテストデータとして用いて性能を評価する（図15）。この手順にしたがい、50個の正常なデータを用いて学習を行った後、人毛を含むブロックを用いて検出性能の評価を行った。MLF-SCを用いて画素ごとに再構成誤差

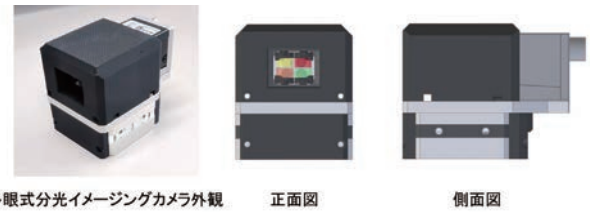

図14　多眼式分光イメージングカメラ

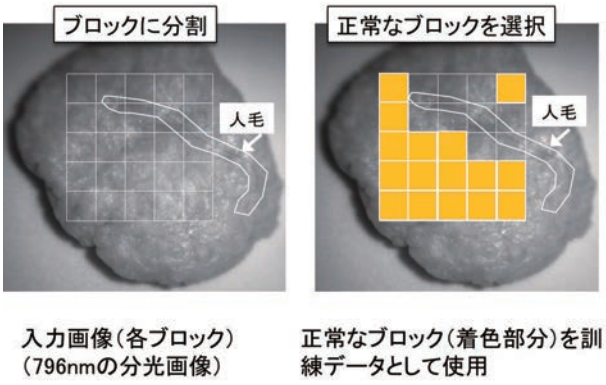

図15　豚挽肉の訓練データの設定
（異物を含まない、正常ブロックのみを用いて学習）

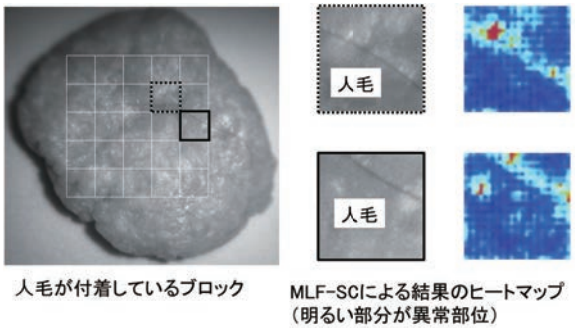

図16　MLF-SCによる豚挽肉の人毛の検出結果
（正常部分は暗く、異物部分は明るく表示）

を計算し、結果をヒートマップで表示した（図16）。異常がある部分は再構成誤差が大きくなる、すなわち、ヒートマップで明るく表示される部分の位置が人毛部分と一致することから、MLF-SCを用いることで異物検出が可能であることが確認できた。

6. リザバーコンピューティングによる異常検知技術

リザバーコンピューティングを用いて時系列データ向けの異常検知技術の開発を行った。3.2節で述べたように、正常音を正しく予測することができれば、正常音とは異なる異常を検知することが可能となることから、ここではリザバーコンピューティングの実装の一つであるESNを用いて、工場の騒音を正常音としたときに正しく予測可能かを試みた。

6.1　ESNの概要

リザバーコンピューティングには様々な実装が提案されているが、本研究では、図17に示すようなRNNをベースとしたESNを用いた。RNNは時系列データを扱えるように出力データを入力側のネットワークに戻す構造のニューラルネットワークモデルであり、音声認識や自然言語処理などへ応用されている。典型的なRNNでは中間層（隠れ層）が循環構造となっており、時間的に過去の情報を保持しながら現在の入力と組み合わせて学習を行うことで時系列データのパターンを学習することができる。リザバーコンピューティングは中間層のニューロンの結合がランダム、かつ結合の重みが固定されているRNNである。出力層の重みのみを学習すればよいため、計算量は少なくなり高速化を実現できる。

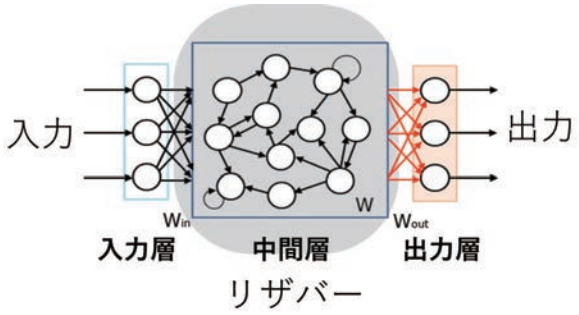

図17　ESNのネットワーク構造

6.2　ESNによる学習

多くの工場には、搬送や加工のための装置が多数設置されているが、外観から検知できない装置の異常はその動作音から検知されることが多く、長年の経験を積んだ熟練作業者が診断している。工場では複数の装置の動作音が重なり合い、工場内全体の騒音となっている。すべての装置が異常なく動作している時の騒音を予測するモデルを構築し、異常発生時

の騒音との差を評価することで異常を検知することができる。ここでは、ESNを用いて工場騒音を予測するモデルの構築を行った。具体的には、工場騒音を１次元の音響信号としてESNに入力し、出力層の重みはリッジ回帰を用いて学習を行った。

6.3　工場騒音の予測

工場内においてサンプリング周波数48kHzで録音した5000ステップ（データ長約104ミリ秒）の騒音データのうち、2500ステップ（データ長約52ミリ秒）のデータを使用して学習モデルを構築した。当該モデルを使用して2501ステップ～5000ステップのデータを予測したところ、音の強度に差はあるが周期のずれがない信号を予測することができた（図18）。この結果から、本学習モデルにより工場騒音を予測でき、工場騒音と異なる音（異常音）を判別可能であることを確認した。ただし、ネットワークの重みを調整するための学習期間において、背景音が不安定な非定常音である場合や、検出対象となる異常音の周波数やデータ長、背景音との強度差なども予測性能に大きく影響するため、ハイパーパラメータの調整や訓練データのデータ長などを十分調整する必要がある。

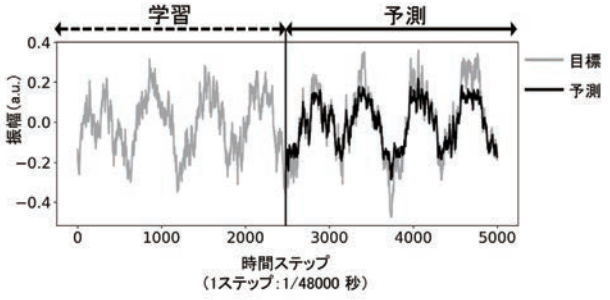

図18　リザバーコンピューティングによる工場騒音の予測

7. おわりに

エッジデバイス上で学習・推論の機能を実現することを目指して、少数の訓練データのみの学習で十分な予測性能を実現可能なスパースモデリングやリザバーコンピューティングによる異常検知技術の開発を行った。スパースモデリングではVSCおよびMLF-SCの２通りの手法についてPythonでコードを実装し、MNIST手書き文字の認識や食品に混入する人毛などの異物の検出に適用して性能評価を行った結果、スパースモデリングを使用しない従来の方法に比較して少ない訓練データで学習モデルを構築でき、さらにその性能が向上することを明らかにした。また、リザバーコンピューティングではESNをPythonでコードを実装し、50ミリ秒程度の訓練データで工場騒音の予測が可能であることを確認した。

今後、半導体の設計技術や製造技術がさらに発展し、最先端の計算機がクラウドサーバーに使われる一方で、工場で稼働するIoTシステムや自動車のADAS（Advanced Driver-Assistance Systems）などでは電力効率のよい小規模なエッジデバイスが使われ続けるものと考える。人手不足を解消するための自動化、省人化のためにAIを実装するニーズはますます増加していくことから、本研究で取り上げた小規模なエッジデバイスにAIを実装する技術はさらに重要になっていくものと考える。今後、改良を進め、当該技術の適用事例を広げるなど普及に向けた取り組みを進めていく。

参考文献

1）Diederik P. Kingma, Max Welling: *arXiv*:1312.6114v1, *Published as a conference paper at ICLR2014*

2）Francesco Tonolini, Bjørn Sand Jensen, et al.: *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference, PMLR* 115:690-700, 2020

3）Karen Simonyan, Andrew Zisserman: *arXiv*:1409.1556v6, *Published as a conference paper at ICLR 2015*

4）Ryuji Imamura, Kohei Azuma, et al.: *arXiv*:2104.04289 (2021)

5）H.Jaeger: *German National Research Center for Information Technology GMD Technical Report*, Vol.148, 34, (2001)

6）Yann LeCun, Leon Bottou,et al.: *Proc. Of the IEEE*, Nov. 1998

7）本間稔規，岡崎伸哉，他：北海道立総合研究機構工業試験場報告，No.321, (2022)