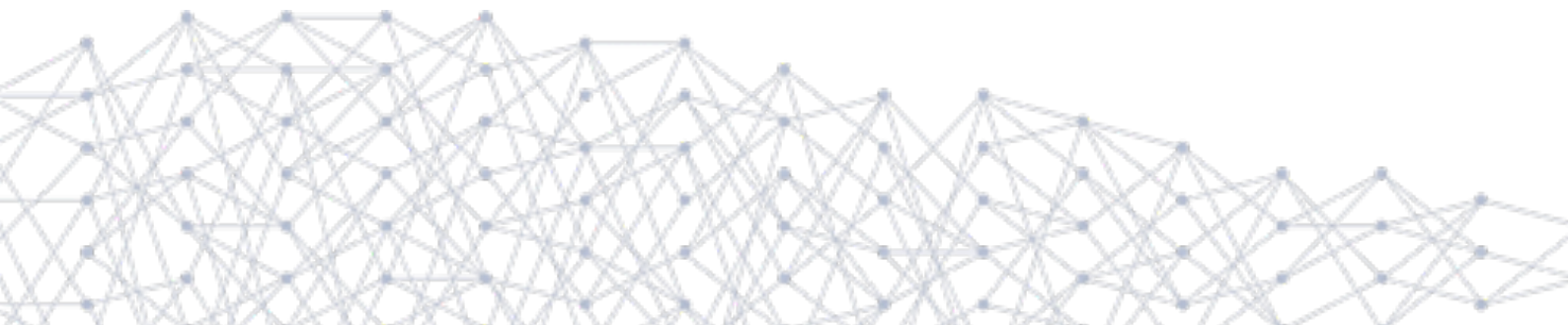


# 縫製位置検出のための ディープラーニングモデル検討

2019 年 6 月 28 日（金）



## 目次

<b>1</b>	<b>本プロジェクトの目的</b>	<b>1</b>
1.1	背景	1
1.2	Deep Learning とは	1
1.3	目的	2
1.4	ロードマップ	3
<b>2</b>	<b>Deep Learning による縫製位置検出方法</b>	<b>4</b>
2.1	画像からの縫製位置検出方法	4
2.2	Segmentation 手法	4
2.3	量子化ネットワーク	6
<b>3</b>	<b>データセット・学習手順</b>	<b>7</b>
3.1	アノテーション	7
3.2	学習データセット	7
<b>4</b>	<b>モデル作成と評価指標</b>	<b>9</b>
4.1	Segmentation モデル	9
4.2	学習・チューニング	9
4.3	評価指標	9
<b>5</b>	<b>評価結果</b>	<b>11</b>
5.1	学習・推論結果	11
5.2	ハードウェアへの実装評価	12
<b>6</b>	<b>全体を通してのまとめ</b>	<b>14</b>

# 1 本プロジェクトの目的

## 1.1 背景

人・モノ・組織など、あらゆるものをつなげる ICT・IoT の世界的な流れは、1つの業界に留まらずあらゆる産業において多大な影響を及ぼしつつある。それらは、既存プレイヤーの役割や主導権のシフト、異業種からの新規参入、新たな業態創出といった変革を促し、これまでにない産業・市場・投資・価値を産み出すだけでなく、既存の構造を破壊・再生さえも引き起こしている\*<sup>1</sup>。例えば製造業においては、旧来の垂直統合型・水平分業型から、データドリブン型のプラットフォーム構造やオープンイノベーションといった新しい枠組みが産まれるとともに、2017 年度には我が国の在り方として「Connected Industries」のコンセプトが発信されるなど、変革が加速する素地が整いつつある\*<sup>2</sup>。

繊維産業に目を向けてみると、世界的には成長を維持しているものの、国内では衣類単価の減少・人材不足が、海外では人件費の高騰といった課題がある\*<sup>3</sup>。この課題に対し、高付加価値素材への注力に加え、JUKI 株式会社によるスマートファクトリー構想をはじめとした ICT の活用による新しい価値創造が進められており、「モノ」を作り売る産業から、モノから新たな「体験・価値」を創る産業へと移りつつある\*<sup>4</sup>。

製造業における新たな「体験・価値」の1つとしては、製造工程から人を排除した完全なる自動化が考えられる。第一次産業革命以来、人の手によりモノの製造・検査・配送等が行われていたが、制御工学・工作機械の発展により、近年では製造工程の一部に人が関わらない自動化が図られている。この領域に、さらに人工知能・ロボティクス・IoT といった分野が加わり、人が知と経験を以って行う必要がある部分さえも機械に置き換わろうとしている。人を介さないことの価値は、人件費や人材確保などの側面に加え、品質を安定させること・時間/場所/環境を問わないこと、そして人がより生産性の高い業務に専念できることなどが挙げられる。この自動化を実現する手段の1つとして、人工知能分野では“Deep Learning”技術の研究開発が行われている。

## 1.2 Deep Learning とは

Deep Learning とは、人の脳が持つ神経ネットワークの一部分と類似した計算アルゴリズムであり、コンピュータ自身が処理のルールを学習できるものである。旧来の制御工学においては、人がルールを定義しコンピュータにプログラムしていた。このルールは非常に堅く・かつ人が無理なく設定できる数量であるため、実際の環境において発生しうる揺らぎにより定義されていない事象がプログラムに入力された場合、プログラムではその入力を適切に処理できず、予期せぬ動作・エラーが発生してしまう。一方、Deep Learning では、バリエーションのある実データを使い、人がプログラムできない程の多数のパラメータ・ルールを“学習”させることで、人が未知の事象に対し経験を以って判別するように曖昧さを吸収するルールをコンピュータが作り出すことができる。

その違いとして、写真の中からペットボトルを見つけるプログラムを作る場合を考える。旧来のやり方では、図のようにペットボトルの特徴を定義し、それに合致する写真を選択する手法を取るであろう。例えば、

---

\*<sup>1</sup> 総務省 平成 30 年版情報通信白書より

\*<sup>2</sup> 経済産業省 製造業をめぐる現状と政策課題 Connected Industries の深化

\*<sup>3</sup> 経済産業省 繊維産業の課題と経済産業省の取組

\*<sup>4</sup> 「Value up2022」 2019-2021 年度の中期経営計画

大きさは 280ml, 350ml, 500ml, 1L, 1.5L, 2L とし、キャップは円形で形は四角柱や六角柱で、カバーの色は... と、かなりのパラメータを検討し定義づけていく。しかし、仮にそれらを入力できたとして、ペットボトルを斜めから撮影している・人が手に持っている・新発売の特殊な形状である... といった、定義から外れる条件下では、ペットボトルを認識できないであろう。仮にそれらをプログラムで定義しようとしても、データのバリエーションが無数に存在するため、実現することは非現実的である。一方で、Deep Learning では、ペットボトルの持つ特徴量を学習を通じてコンピュータ自身がルール化しアルゴリズム内のパラメータとして記憶する。その結果、仮に学習していないペットボトルが写真に含まれていたとしても、その特徴量が学習内容とある程度一致するものであれば認識することが可能となる。この判断は、人の認識技術と近い振る舞いであり、曖昧な・想定と異なる状況が発生したとしても適正に対処できうる可能性があるため、人の経験などで行っていた業務を Deep Learning を使ったコンピュータに置き換えることができる。

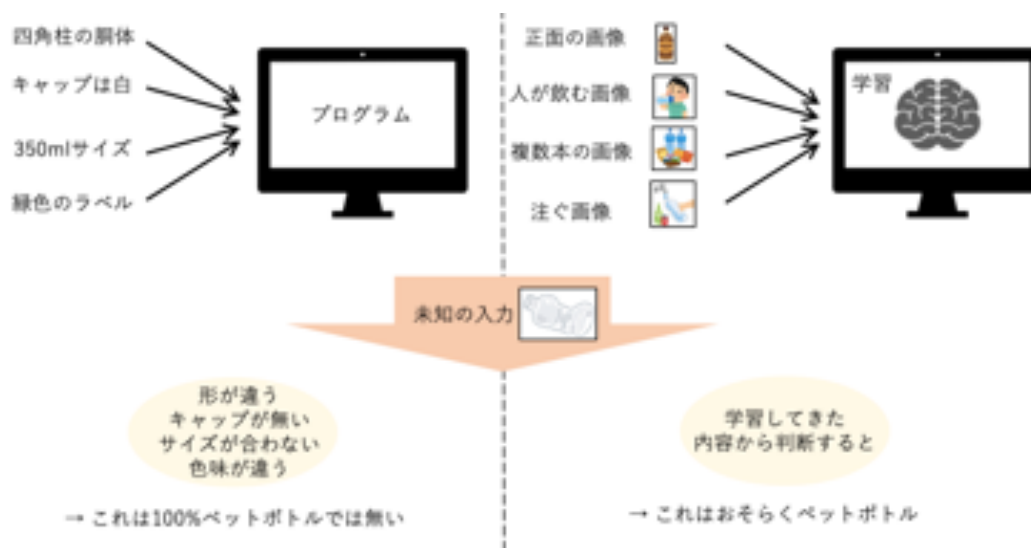


図1 旧来のプログラムと Deep Learning の違い

### 1.3 目的

本プロジェクトは、紡織機器・産業用ミシンにおいて、縫製自動化のために Deep Learning が適用できるか予備検討を行うものである。ミシンにおいて縫製位置を自動で設定するには、理想的には CAD など設計された縫製図面を使用すれば良いと考えられるが、実際には、縫製途中において布地に歪みや位置ずれが発生するため、図面から算出された理想的な縫製位置とは異なる場所へ最適位置が移動してしまう。この位置ずれは、縫製生地や生地の縫い場所などにより無数のパターンが発生し、旧来のプログラム手法のように、位置ずれのルールを定義するやり方では全てのケースを満足することができない。そこで、縫製時の写真を学習データに用いた Deep Learning モデルを作り、未知のデータ・位置ずれが発生した場合でも検出できるか検証を行う。

## 1.4 ロードマップ

本プロジェクトにおける開発ロードマップのイメージを図に示す。紡織機器を自動化させるためには、Deep Learning モデルの作成に加え、モデルを搭載するハードウェアの選定/開発/検証、カメラや紡織機器との接続検証、紡織工程や紡織機器製造工程を考慮した運用設計といった、システム全体の設計/開発/検証が必要となる。このうち Deep Learning 部分では、モデルの精度・速度とハードウェア性能にトレードオフ関係があることに加え、外付けハードウェア機器を使うのか・既存システムの CPU を使うかなど、システム構成も大きく変わるため、システム設計とモデル開発を並行して実施する必要がある。今回の検討フェーズでは、本プロジェクトにおける Deep Learning 適用可否の評価を主として行うが、製品化・サービス化までの工程を検討し、着実に進めていく必要があると考えられる。

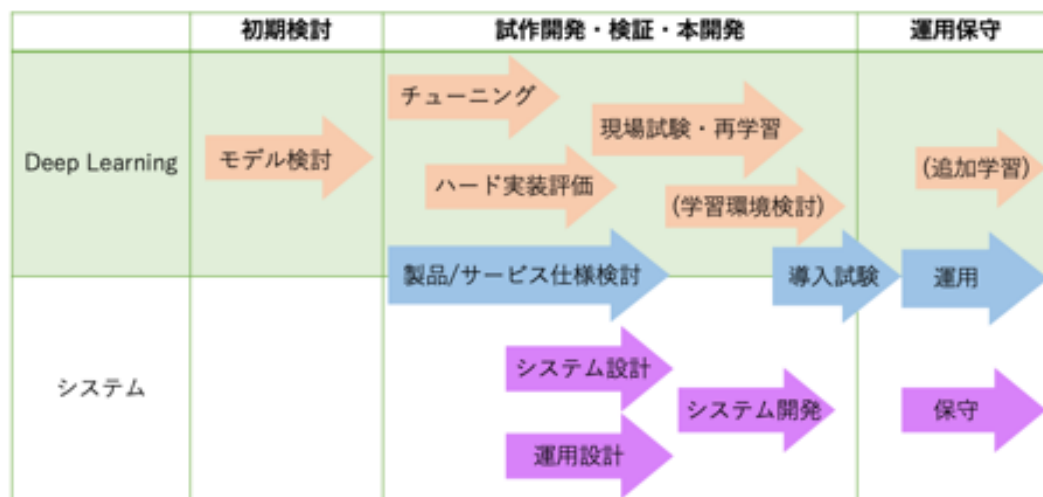


図2 開発ロードマップ

## 2 Deep Learning による縫製位置検出方法

### 2.1 画像からの縫製位置検出方法

学習の対象となる画像例を図3に示す。この画像は、製造工程における縫製布の表面を拡大したものであり、点群のようなものは、布地表面の模様である。ここで、縫製位置は点群の中点を結んだ図4であり、このXを描くラインを自動で検出することが目的となる。例示した画像は比較の見やすいが、実際には後述するように、縫い跡や凹凸による影の発生などにより、様々な画像パターンが存在する。これらの異なるパターン全てに対応するためには、Deep Learning を使った位置検出を検討する意義が高いと言える。尚、実際にはX以外の形状・曲面を描くパターンもあるが、今回は学習の対象外とする。

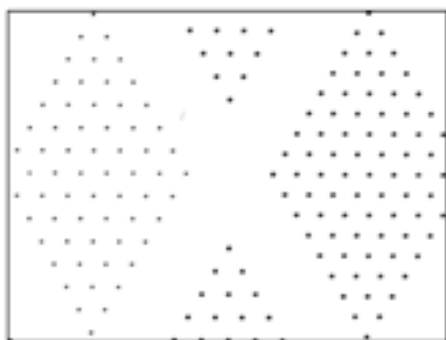


図3 学習画像

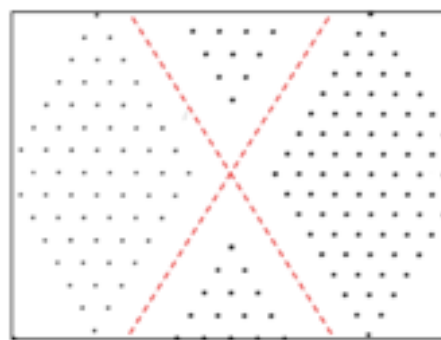


図4 縫製位置

Deep Learning による位置検出方法について検討を行う。縫製位置は、点群のエッジ部分からの中点に位置する。このため、例えば図5のように、点群の領域や、点群のエッジ部分を Deep Learning により自動で検出することができれば、図6のように、その中間を画像処理により抽出することは容易に実現しうるものである。この点群領域を検出するため、今回は Deep Learning を用いた Semantic Segmentation 手法を用いて検討する。

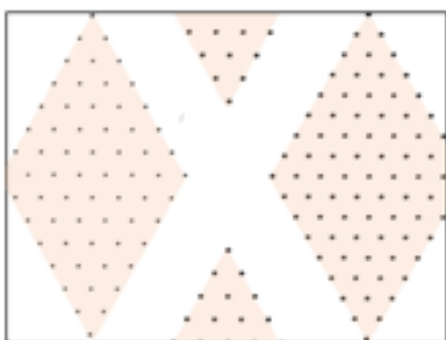


図5 点群領域の推定

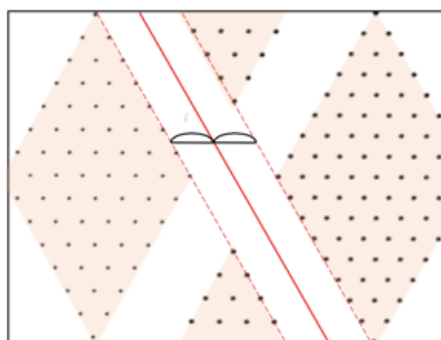


図6 縫製位置の検出

## 2.2 Segmentation 手法

Semantic Segmentation とは、画像の特徴を画素レベルで把握する手法のことである。その実現方法は多数あるが、Deep Learning を使った手法として SegNet<sup>\*5</sup>について検討を行う。

SegNet とは、ケンブリッジ大学の研究グループにて提案された、画像をある画素単位で分割し、その特徴量を算出することで物体の種類に応じた領域の検出が可能となるオープンソースの Deep Learning アーキテクチャである。その構造を図 7 に示す<sup>\*6</sup>。SegNet は、Encoder と Decoder の処理ネットワークに分割できる。Encoder は、VGG16 モデルをベースとした画像の局所特徴を抽出するための畳み込み層<sup>\*7</sup>、ダウンサンプリングを行い画素単位を細かくしていくプーリング層、計算途上で勾配消失を防ぐ Batch Normalization 層などから構成されており、細分化した画素値から物体の持つ特徴量を学習していく。Decoder では、Encoder によって得られた物体の種類と大まかな位置情報を持つ特徴量マップに対し、アップサンプリングと畳み込み処理を行うことで元の解像度へ対応づけを行っていく。ここで、図 8 に示すように、アップサンプリングでは Encoder ネットワークにおいて予め記憶したプーリング層の画素関係・Pooling Indices を使い、より適した位置に特徴量を割り当てる。例えば、c を割り当てる際にプーリング層では図のような特徴量を持っていたとすると、割り当てられるべきは左下の領域となる。割り当てた特徴量は畳み込み演算により滑らかになり、最終段において特徴量を際立たせる Softmax 処理を行うことで、入力画像の特徴を画素単位で表示することができる。図中の例においては、道路の風景写真に対して、道路・路側帯・車といった物体の特徴をマップ化することで異なる色で塗り分けている。

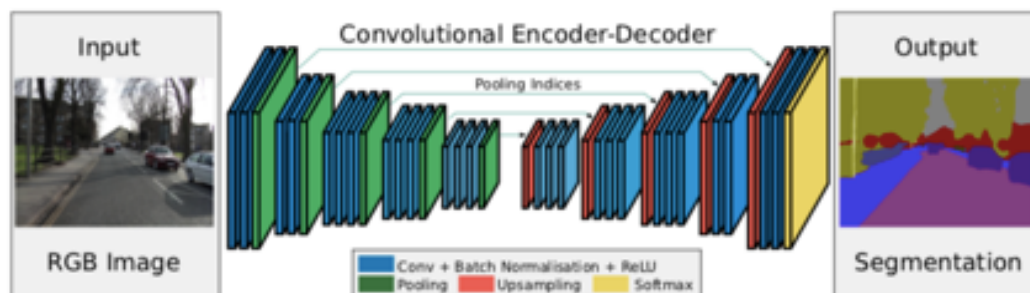


図 7 SegNet アーキテクチャ

<sup>\*5</sup> <http://mi.eng.cam.ac.uk/projects/segnet>

<sup>\*6</sup> Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," <https://arxiv.org/pdf/1511.00561.pdf>

<sup>\*7</sup> Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," <https://arxiv.org/pdf/1409.1556.pdf>

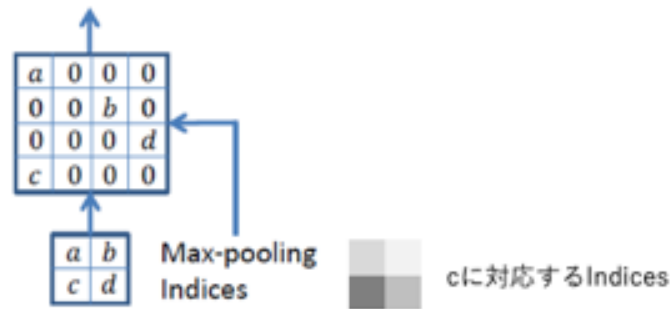


図 8 Upsample 処理

## 2.3 量子化ネットワーク

LeapMind では、Deep Learning モデルを軽量なエッジデバイスで動作させるため、モデルの量子化を行っている。量子化とは、通常 32bit や 16bit の浮動小数点で表現されている重みや特徴量データを、より少ない bit 数の固定小数や整数などで表現するものである。畳み込みを行う Deep Learning モデルでは、演算量の大部分を畳み込み処理、即ち乗算と加算が占めている。この演算過程において、処理すべき単位が高次の浮動小数点であった場合、必要な回路規模は固定小数点や整数と比べると格段に大きくなるため、消費電力やメモリ領域の増加、処理速度の低下といった問題を引き起こす。特に演算回路規模の少ないエッジデバイスにおいては、性能劣化だけでなく、そもそもハードウェアへの実装ができないケースも発生しうる。このため、精度劣化がさほど発生しない範囲内において、重みや特徴量データをより少ない bit で表現する量子化が有効な解決策となる。LeapMind では、特にこの量子化に対して技術的な優位性があり、精度劣化を抑えつつ内部での処理を最小 1bit で実施することができる。今回は、SegNet をベースとし量子化技術を加えたモデルを用いて学習・検証を実施する。



## 3 データセット・学習手順

### 3.1 アノテーション

Segmentation では、教師あり学習を実施する。即ち、学習に使用する画像に対して、予め検出したい領域とそのカテゴリを用意する必要がある。画像に対し注釈をつけるという意味で、この作業をアノテーションと呼ぶ。アノテーションを行うためのツールとして、Microsoft 社の VoTT \*8を使用した。

実際の学習画像に対するアノテーション例を図9に示す。縫製位置は画像中心部分のXを描く空白部分であるが、画像の両脇にも見切れた形でXが残っている。この部分がモデルの精度に影響する可能性があることを考慮して、今回はこの見切れた部分についてもアノテーションを実施した。また、領域の線は、Xに隣接する布地のドット模様の中心を通る形としている。

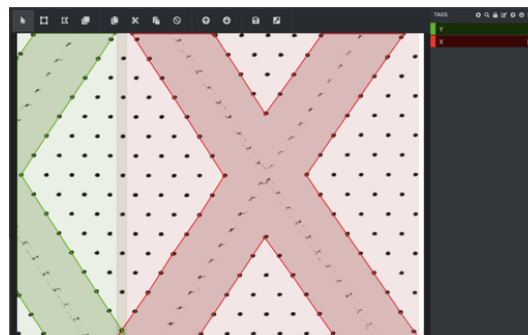


図9 アノテーション例

### 3.2 学習データセット

Segmentation モデルを作成するために使用した学習データを記述する。縫製工程において撮影された布地の写真は、その特徴から”Normal” ”糸残り” ”縫い跡・影あり” ”縫い跡あり” の4つに分類される。それぞれの特徴は以下の通りである。

- ・ Normal : 生地表面の模様以外には写り込みが無い写真
- ・ 糸残り : 中心部に縫製糸が残り見えている写真
- ・ 縫い跡あり : 中心部に縫製跡が残っている写真
- ・ 縫い跡・影あり : 中心部に縫製跡が残っており、かつ生地表面の凹凸による影が写り込んでいる写真

実験では、それぞれの特徴を持った写真が一定数以上入るデータセットを用いてモデルの学習を行うとともに、学習データに含まれない画像を用いて推論を行い、モデルの評価を行った。この条件下で実験したものを“実験1”と定義する。また、実験1の結果として苦手とする特徴が発生した場合、その特徴を持つ画像を実

\*8 VoTT : <https://github.com/microsoft/VoTT>

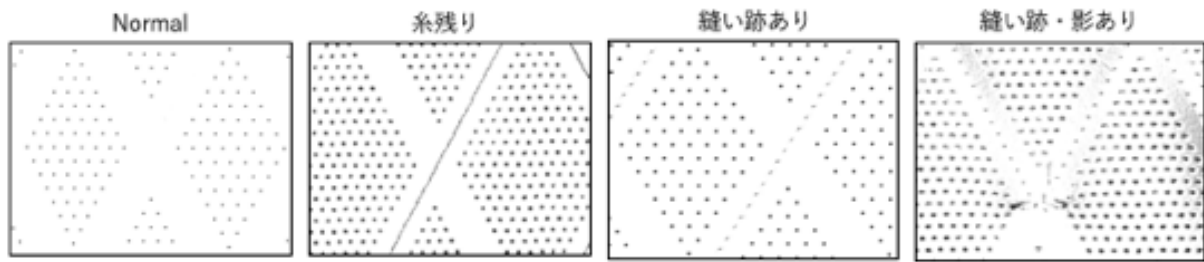


図 10 写真の特徴

験 1 で使用したデータセットに追加して再度実験を行うこととした。後者を“実験 2”、“実験 3”とし  
実験 1 では、受領した約 1 8 0 0 枚の中からそれぞれの特徴が最低 2 0 枚以上含まれる形で 1 6 1 枚を抜粋し、データセットとして学習に使用した。実験 2 では、実験 1 の結果次第で苦手な特徴に対し 1 0 枚の追加を、実験 3 では 2 0 枚の追加を行うこととする。

- ・ Normal : 4 0 枚
- ・ 糸残り : 2 6 枚
- ・ 縫い跡あり : 7 4 枚
- ・ 縫い跡・影あり: 2 2 枚

評価用のデータは、下記の内訳で 3 7 枚を使用した。同じ特徴内においても写真にバリエーションがあるため、ある程度のパタンの差異が出るようにデータを選択している。Normal では比較的画像のバリエーションが少ないため、評価用データの枚数を他に比べ半分程度の分量としている。

- ・ Normal : 6 枚
- ・ 糸残り : 1 1 枚
- ・ 縫い跡あり : 1 0 枚
- ・ 縫い跡・影あり: 1 0 枚

一般的に Deep Learning モデルを作成する場合、最も重要な要素の 1 つが、学習データの質と、質の高いデータの量である。今回のケースでは、学習に使う写真が実際の使用状況に即したものであり、複雑性があまりなく、かつある程度のバリエーションがあるため、質の面では一定の水準には達していると考えられる。学習・評価に使用したデータは、別途 DVD に収めて提出する。

## 4 モデル作成と評価指標

### 4.1 Segmentation モデル

Segmentation モデルは、前述した通り、SegNet に量子化技術などを加えた LeapMind 独自のものを使用している。これは“blueoil”に組み込まれているモデル“LmSegnetV1Quantize”と同等の構成である<sup>\*9\*10</sup>。blueoil とは、LeapMind が持つ技術の粋を集めた組み込み機器向け Deep Learning のオープンソース・開発プラットフォームである。本プラットフォームで開発されたモデルは、FPGA, CPU, GPU など、様々なデバイスへと実装が可能となっており、その機能は更に日々アップグレードされ続けている。

### 4.2 学習・チューニング

学習時における代表的なパラメーターとして以下を設定し用いた。

バッチサイズ	: 32
エポック数	: 500
初期学習率	: 0.001
学習率変化	: 3-step-decay
イメージサイズ	: 128 × 128
オーギュメンテーション	: Blue, Brightness, Color, Contrast, FlipLeftRight, FlipTopBottom

バッチサイズとは、データセットをいくつかのサブセットに分けて 1 回学習する際のサブセット 1 つに含まれるデータ数であり、エポック数は、全てのデータセットが学習に使われる回数を示す。例えばデータセットが 320 枚の場合、上記の設定値では、1 回の学習には 32 枚の画像が使われる。全てのデータセットが学習に 1 回使われるまでの学習回数は 10 回で、設定されたエポック数より総学習回数が 5000 回となる。学習率とは、学習時における誤差パラメータの変更量に相当し、学習工程の 1/3 毎に学習率を 1/10 にする手法を取っている。イメージサイズは入力画像の画素数であり、受領した元画像のサイズ 1280 × 960 を小さく・正方形にリサイズしている。

オーギュメンテーション・データ拡張とは、元の学習データに対して画像処理を加え、新たな画像を作成することである。この処理自体では、データそのものの特徴量を増やすという効果はないものの、例えば検出したい物体が現実世界とは異なり左寄りになっている・昼間の画像が多い・障害物がなく綺麗に写りすぎている、といった条件において、データの偏りを緩和する効果などがある。今回は、製造工程での使用を念頭に入れ、主に画像の反転や輝度・色味の変化のみを実施している。

### 4.3 評価指標

Segmentation における学習結果の評価指標として、モデルにより予測された結果と、対応する教師データのアノテーション結果との比較を行い、検出すべき対象部位の重なり具合・IoU (Intersection over Union)

---

<sup>\*9</sup> blueoil : <https://blue-oil.org>

<sup>\*10</sup> blueoil : <https://leapmind.io/news/content/2460>

を算出する。図のような例であれば、教師データと検出データで結果が一致している領域は 5.0 に対し、全ての領域が 13 であるため、約 0.38 となる。この指標においては、教師データと検出データが完全に一致していれば IoU は 1.0 となるが、例えば教師データの領域内に検出データが全て含まれている・または検出データの領域内に教師データが全て含まれているような状況においても、教師データ・検出データ双方が一致しない部分が少しでもあれば、結果は 1.0 より小さくなる。このため、IoU は高ければ高いほどモデルの精度は良いと言えるものの、現実問題として完全に 1.0 となるケースはあまりなく、また、仮に結果が 1.0 より低い状況であったとしても実使用上問題が発生するかどうかは実際の推論結果を確認し分析することで判断していくこととなる。今回は、IoU と推論結果兩者について評価を実施した。

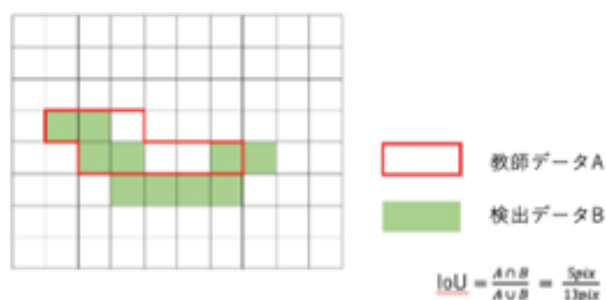


図 11 IoU 計算方法

## 5 評価結果

### 5.1 学習・推論結果

実験1の学習結果を図12に示す。学習ステップに対するIoUの値は、400ステップ程で高い水準へと収束し、高い水準を維持したまま収束していることがわかる。この結果は学習が問題なく進んでいることを示しており、本取り組みにおけるDeep Learningの適応可能性は十分にあると考えられる。このモデルに対し、推論データを適用し評価を行った。推論結果の一覧は付録に掲載する。概ね所望の領域が十分検出できている一方、図13に示すように“糸残り”のデータに対しては、糸で囲まれた部分を検出領域として誤判定してしまうケースが見られている。このため、苦手な特徴を持つ“糸残り”のデータを学習に追加することで、検出結果の改善ができるか検討を行った。実験2では、実験1の学習データに“糸残り”の特徴を有する画像を10枚追加し、実験3においては、実験2に対し更に20枚追加し、モデルを作成し評価を行った。

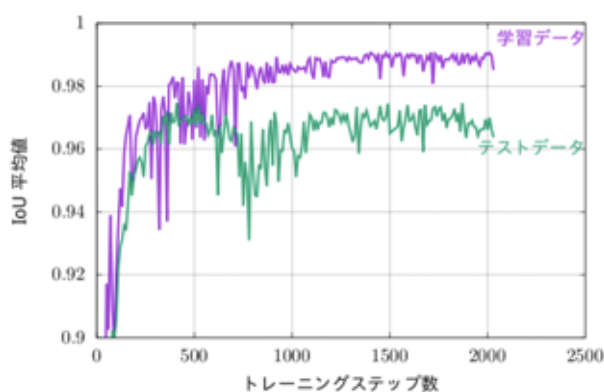


図12 実験1：トレーニングステップに対するIoU

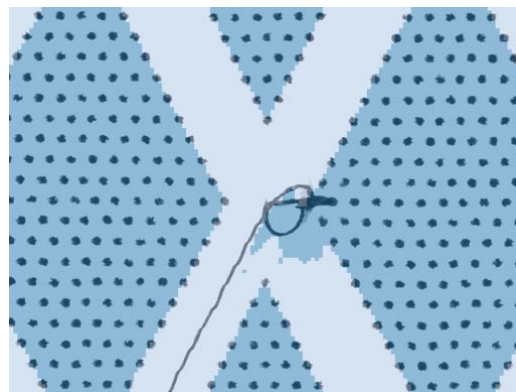


図13 実験1：“糸残り”のデータに対する推論結果

実験2, 実験3のIoUならびに“糸残り”の特徴に対する推論結果を図に示す。実験1と実験2を比較すると、実験2におけるテストデータのIoUがより滑らかに収束していることが分かる。Deep Learningを使ったシステムを実用化するには、学習データに含まれない未知のデータに対する推論結果の性能、即ち汎化性能が重要となる。このため、実験2の方が実験1に比べ、より汎化性能が高い・より良いモデルとなったと言える。一方、実験2と実験3を比較すると、実験3における学習データのIoUが揺らいでいる。これは、苦手なデータが増えたことによる影響と考えられる。しかし、テストデータの結果を比べてみると、実験2と実験3で差がなく、モデルとしてみるとより安定性が増している可能性が高い。実際、“糸残り”の特徴に対する推論結果を比べてみると、これまで発生していた誤り部分に改善が見られていることから、より汎化性能が高いモデルができたと考えられる。また、画像のエッジ部分に凹凸が見られるが、この部分はSegmentationにおいて画像の縮小・拡大を行なっているためであり、スムージングなどの後処理により平滑化が可能となる。

尚、今回は少量の学習データにより実験を行っているが、実使用環境において汎化性能を向上させるには、質の高い学習データの量を増やすことに加え、学習時のパラメータである学習率やバッチサイズなどのチューニングを行うことが必要となる。

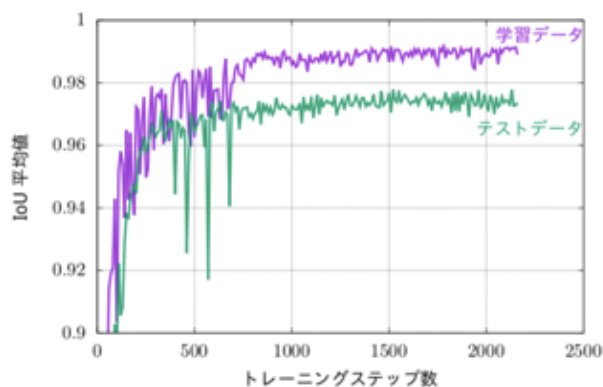


図 14 実験 2 : トレーニングステップに対する IoU

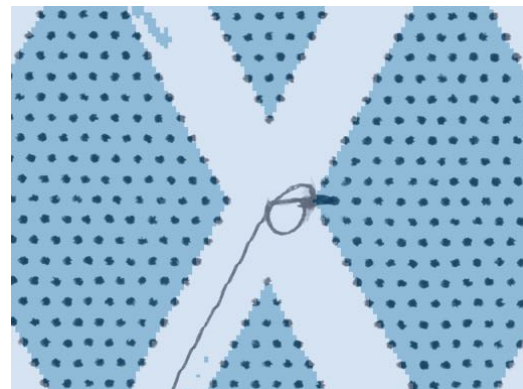


図 15 実験 2 : “糸残り” のデータに対する推論結果

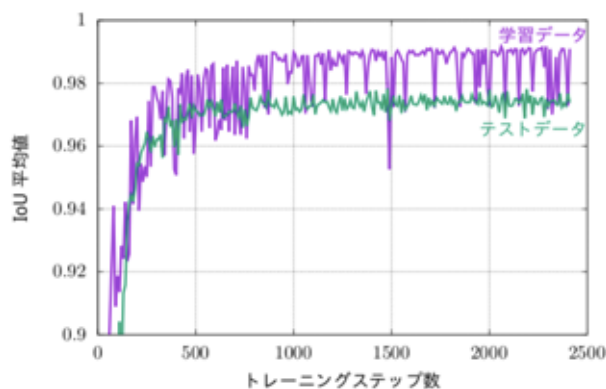


図 16 実験 3 : トレーニングステップに対する IoU

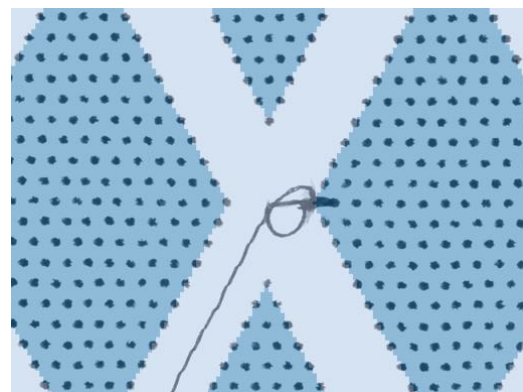


図 17 実験 3 : “糸残り” のデータに対する推論結果

## 5.2 ハードウェアへの実装評価

作成したモデルをハードウェアに実装し、処理速度について評価を行った。今回使用するハードウェアとしては、Terasic 社により販売されている DE10-Nano を使用した<sup>\*11</sup>。外観ならびに構成を図に示す。DE10-Nano には、Intel 製のローエンド FPGA である CycloneV SE<sup>\*12</sup>が搭載されている。CycloneV SE には、同一パッケージ内に 28nm プロセスで作成された FPGA とデュアルコアの ARM Cortex-A9 が搭載されており、ARM プロセッサ側で Linux OS が動作する構成となっている。Deep Learning モデルを実装する際には、畳み込み演算などの演算処理を FPGA 側で行い、画像のリサイズや特徴量の配置といった部分を ARM 側で行なっている。処理速度の評価では、DE10-Nano ボード内に保存した画像に対し推論処理を 100 回実施し、推論結果が得られるまでの時間を計測した後、単位時間に推論処理ができる画像枚数・FPS の算出を行った。今回のモデルを実装し評価した結果、2.76FPS (1 秒間に 2.76 枚の画像推論処理が可能) が得られた。処理速度は、モデルの層構造・チャンネルを調整することで改善が可能である。また、今回の推論画像は入力サイズを  $128 \times 128$  pixel としているが、単純計算で画像サイズを  $96 \times 96$  pixel とすると処理速度は 2 倍とな

<sup>\*11</sup> DE10-Nano : <https://DE10-Nano.terasic.com>

<sup>\*12</sup> Cyclone V SE : <https://www.intel.co.jp/content/www/jp/ja/products/programmable/fpga/cyclone-v.html>

る。一方で、推論する画像の画素数が下がるため、精度劣化が生じる可能性がある。

実運用においては、カメラ I/F における撮像～通信や、推論結果を縫製装置に戻すための電文作成・処理・通信といった部分が更に上乗せされるため、必要となる処理性能の算出と性能確保、ハードウェアの選定・コスト算出など、製品化・サービス化へ向けロードマップを踏まえた検討が必要になると考えられる。

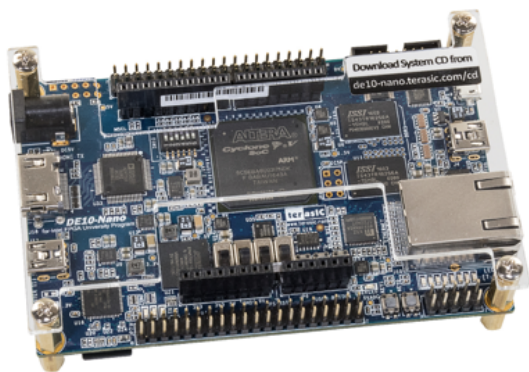


図 18 DE10-Nano ボード・外観写真

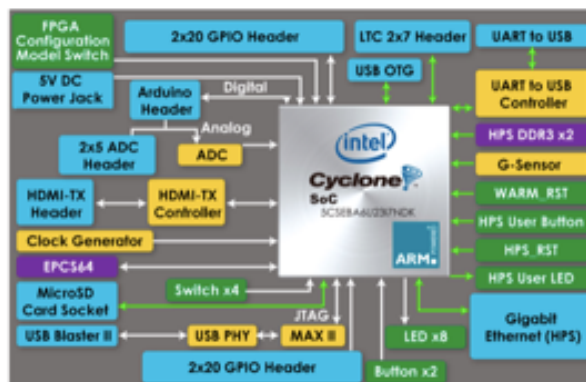


図 19 DE10-Nano ボード・構成

## 6 全体を通してのまとめ

縫製の自動化へ向けて、縫製時の写真を学習データに用いた Deep Learning モデルを作り評価を行った。その結果、高い IoU と実データでの推論評価より、Deep Learning により縫製位置を検出することができる可能性が高いという結論に至った。本モデルを搭載した自動縫製装置の実用化には、モデルの精度や処理速度向上のためのデータセット収集・チューニングに加え、CPU・軽量 GPU といった実装するハードウェアの検討、製品・サービスの仕様検討、システム・運用設計といった部分が必要となる。本結果はその第一歩を踏み出すに十分な結果を得たと言える。



## 付録：推論結果一覧

ここでは、左の列から順番に実験 1, 2, 3 全ての推論結果を列挙する。

