

CapsNet を用いたウェハ上不良パターンの分類 Wafer Map Defect Pattern Classification using CapsNet

藤田樹[†] 永村美一^{††} 新井雅之[‡] 福本聡^{††}
Itsumi Fujita Yoshikazu Nagamura Masayuki Arai Satoshi Fukumoto

1. はじめに

半導体製造においては、テスト工程で得られる不良に関する情報が信頼性向上のための重要な役割を担っている。特に、各製品チップ（ダイ）のテスト結果をウェハ上の位置に反映したウェハマップの不良パターンを分類（識別）することは、製造プロセスにおける不具合の原因を特定する上で重要な情報をもたらす。しかし、半導体製造で得られるデータは大量化かつ複雑化しているため、人手による解析作業では膨大な時間を要することが問題となっている。近年では、画像解析や AI・機械学習を用いた自動分類が研究されている。

これまでに、画像認識で高い精度を出している CNN を使った分類が多く研究されているが [1][2], CNN ではプーリング層によって情報量が減少することや、オブジェクトの位置変化に対応するために多くの教師データが必要であるという課題がある。その解決策の 1 つに Capsule Network (CapsNet) が提案されている。本研究では 2017 年に提案された CapsNet を用いて、ウェハマップの不良パターンのオープンデータである WM-811K を分類した結果をまとめた。

2. 準備：CNN と CapsNet の概要

2.1. CNN

CNN の主な構成要素として、畳み込み層 (Convolutional Layer) とプーリング層 (Pooling Layer) が挙げられる。畳み込み層では、2次元の入力データに対してカーネルと呼ばれるフィルタを掛け合わせることで、入力データの特徴を抽出することができる。こうして計算されたデータを特徴マップと言う。特徴マップはカーネルの数と同じ次元数で出力される。

プーリング層は入力データを一定の間隔で間引くことでデータサイズを圧縮する機能を持つ。これによって、特徴の位置に多少のズレがあったとしてもその差を無視することができる。これらの畳み込み層やプーリング層を繰り返した後に、前層の全てを線形結合させる全結合層を加えたものが一般的な CNN のモデル構成である。

2.2. Capsule Network (CapsNet)

CNN では複数のチャネルの 2次元配列 (スカラー値) の形で特徴抽出を行う。このスカラーを持つニューロンに変わって、ベクトルで特徴を扱うのがカプセル (Capsule) の基本的なアイデアである。カプセルはその長さで特徴の存在確率を表し、向きや位置などの空間情報もパラメータとして保持する。ネットワークの学習には“ルーティング・バイ・アグリーメント (Routing by Agreement)” [3] を使用する。これによって CapsNet では特徴マップを高い次元で表現することができる。

ルーティング・バイ・アグリーメントは、下位のカプセル層がどのように上位のカプセル層へ情報を伝えるのかを計算する。一般的なネットワークでは ReLU, Sigmoid 関数などの

活性化関数を使用するのに対して、CapsNet ではカプセル j の出力 v_j に Squashing 関数

$$v_i = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \quad (1)$$

が用いられる。

下位層 l のカプセル i の出力 u_i は、重みづけられたベクトル $\hat{u}_{j|i}$ の総和 s_j として上位層 $l+1$ のカプセル j へ伝えられる。これらは両者間の重みを W_{ij} とすると、

$$s_j = \sum_i c_{ij} \hat{u}_{j|i}, \quad \hat{u}_{j|i} = W_{ij} u_i \quad (2)$$

とかける。 c_{ij} はカプセル間の結合係数であり、以下のソフトマックス関数で求められる。

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (3)$$

ここで b_{ij} は累積変数で、学習の過程で内積 $\hat{u}_{j|i} \cdot v_j$ を加えていく。つまり、正しい予測の上位層のカプセルへ貢献する下位層のカプセルほど b_{ij} が大きくなっていき、その間の結合係数を大きくするように学習をする。

以上のようにして、入力データの教師ラベルを表現する最上位層のカプセルの長さが大きくなるように特徴を捉えていく。このときの損失関数はマージンロス (Margin Loss) を使用する。各ラベルに対応するカプセル k について、マージンロスの値 L_k は

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2 \quad (4)$$

と求められる。 T_k は教師ラベルで、入力データに不良パターンがある場合 $T_k = 1$ となり、それ以外は $T_k = 0$ となる。また m^+, m^- は予測の精度が十分であるとする閾値で、 λ は学習初期から全てのベクトルが小さくならないように調整する重みパラメータである。これらの値は $m^+ = 0.9, m^- = 0.1, \lambda = 0.5$ とした。こうして求めたマージンロスの総和を取って CapsNet の損失関数の値となる。

他にも Decoder と呼ばれる入力画像をインスタンス化されたパラメータから再現する全結合モデルを加えることでルーティング性能を底上げする。

3. 実験概要：対象データと提案モデル

本研究での CapsNet の構成は図 1 の通りである。まず特徴抽出のための畳み込み層を 2 層用意した。カーネルサイズは 9×9 、ストライドを 4 に設定し、ゼロパディングを使用している。活性化関数は ReLU 関数を使用している。Capsule を扱う PrimaryCaps 層と DigitCaps 層は文献 [3] と同じ構成であり、最後の出力の数だけウェハマップの故障パターンの分類対象クラス数の 8 に変更している。

ウェハマップの故障パターンの分類性能を評価するためにオープンデータである WM-811K を使用した。図 2 に各ラベ

[†] 東京都立大学 大学院 システムデザイン研究科, Graduate School of System Design, Tokyo Metropolitan University

[‡] 日本大学 生産工学部, College of Industrial Technology, Nihon University

^{††} 東京都立大学 システムデザイン学部, Faculty of System Design, Tokyo Metropolitan University

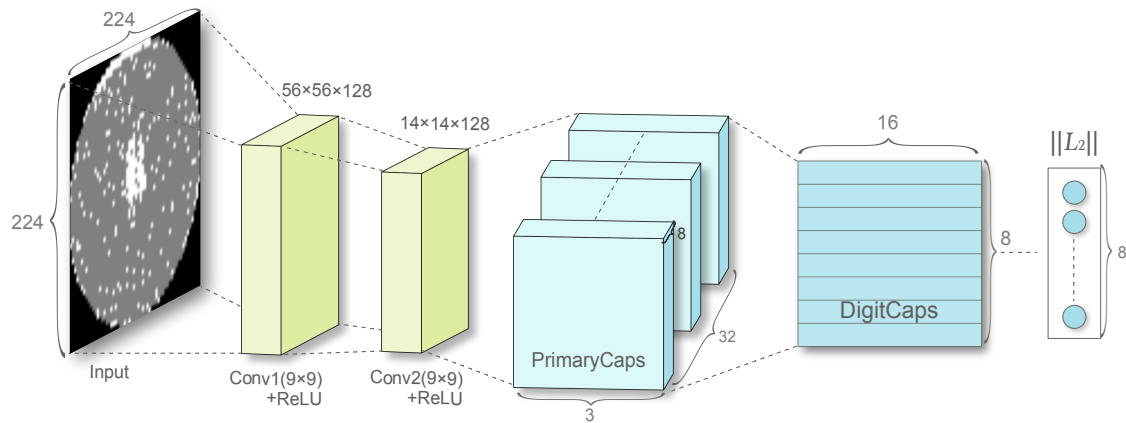


図1 使用した CapsNet の構成

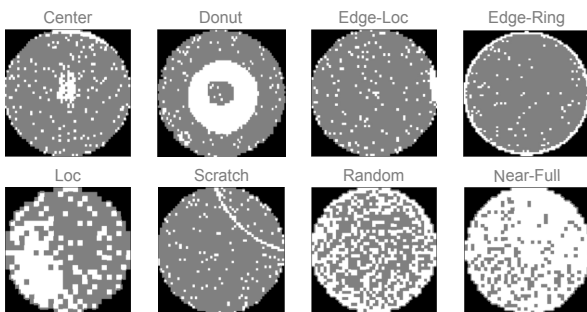


図2 データセットに含まれる不良パターンごとの例

ルの不良パターンのデータの例を示す。白い部分が故障のダイであり、ウェハ上にパターンを作っていることがわかる。

このデータセットは実現場での故障パターンから構成されていて、全データのうち不良パターンのラベルがついているものは約2.5万個である。この各不良パターンのラベル間でデータの個数も異なるため、各ラベルの60%を教師データとして学習に使用し、15%を検証データ、25%をテストデータとした。またデータサイズは、 224×224 に統一した。この処理にはBICUBIC法を使用し、リサイズ後に 9×9 のメディアンフィルタを適用することでデータ上のノイズを軽減させている。

4. 結果

CapsNetによるウェハマップの分類性能と比較するため、同タスクで高スコアを出している文献[1]のCNNを用いて分類性能を事前調査した。表1にCapsNetとCNNでの学習後の結果をまとめた。CNNのLossはCross Entropy Lossを使用し、CapsNetは上述のとおりMargin Lossを使用しているが参考までに載せている。表2ではCapsNetとCNNの各ラベルごとの再現率(Recall)をまとめている。これらの結果から、プーリング層を使用していないCapsNetでもCNNと同等の結果が出せていることがわかる。

また表2に示すように、Loc(局所的に不良箇所が集中するモード)とNear-full(ウェハ上のほぼ全域に不良箇所が存在するモード)以外の6つの不良パターンの再現率が上がっており、特にDonut(中心部に円形に不良が位置するモード)の再現率が向上している。さらに、分類が難しいとされてい

表1 各モデル学習後のテスト結果の比較

| Model | Acc | Loss |
|---------|--------------|---------------|
| CNN | 0.911 | 0.539 |
| CapsNet | 0.925 | 0.0626 |

表2 各テストパターンごとの分類結果

| Pattern | データ数 | Recall | |
|-----------|------|--------------|--------------|
| | | CapsNet | CNN |
| Center | 1074 | 0.980 | 0.970 |
| Donut | 139 | 0.856 | 0.691 |
| Edge-Loc | 1298 | 0.922 | 0.878 |
| Edge-Ring | 2308 | 0.981 | 0.970 |
| Loc | 899 | 0.806 | 0.869 |
| Random | 217 | 0.862 | 0.829 |
| Scratch | 299 | 0.753 | 0.672 |
| Near-full | 38 | 0.895 | 0.895 |
| マクロ平均 | - | 0.882 | 0.847 |

るScratch(線状に連なって不良が位置するモード)でも再現率が向上していることがわかる。一方でEdge-Loc(ウェハの端に局所的に不良箇所が集中するモード)への誤分類が多かったため、不良パターンがどこに位置しているかという認識は今後も課題といえる。

5. おわりに

本研究ではウェハマップの不良パターン分類にCapsNetを使用し、その優位性を確認した。Capsuleを使ったモデルではマルチレベルの優位性が指摘されているため、複数の不良パターンでの認識の検証が今後の課題として挙げられる。また、CapsNet派生の最新モデルの検討なども必要である。

参考文献

- [1] N. Yu, Q. Xu, and H. Wang, "Wafer Defect Pattern Recognition and Analysis Based on Convolutional Neural Network", *IEEE Transactions on Semiconductor Manufacturing*, Vol. 32, No. 4, pp. 566-573, 2019.
- [2] T. Nakazawa and D. V. Kulkarni, "Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network", *IEEE Transactions on Semiconductor Manufacturing*, Vol. 31, No. 2, pp. 309-314, 2018.
- [3] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules", *Advances in Neural Information Processing Systems*, 2017.