

# セマンティックセグメンテーションを用いた 定量的な予測に基づく工業製品の欠陥検出

八木 郁哉† 永田 寅臣† 加藤 博久† 大塚 章正† 越智 裕章‡ 渡辺 桂吾§  
†山口東京理科大学 ‡足利大学 §岡山大学  
E-mail: nagata@rs.soc.u.ac.jp

## 1 背景と目的

製品の検査工程においては未だ目視検査に頼るところが大きく、検査員の作業負荷や検出結果のバラツキなど生産性の低下を招くという問題がある。このような問題を解決するために筆者らも画像認識に特化した深層学習の手法の一つである畳み込みニューラルネットワーク (CNN) を製品の欠陥検出に応用している [1, 2]. 最近では YOLO [3] など一般物体検出用アルゴリズムを用いれば欠陥の場所も特定できるようになってきているものの、傷の長さや面積をもとに定量的な評価に基づき分類したいというニーズには十分に答えることができていないようである。本研究ではこの問題を解決するために画像内の全画素にラベルやカテゴリを関連付けできる深層学習手法であるセマンティックセグメンテーション (SS)[4] を応用し、代表的モデル U-Net により、工業材料の製造工程で発生するひび状の欠陥の検出を試みたので報告する。

## 2 実験内容

セマンティックセグメンテーションは、画像内の全画素にラベルやカテゴリを関連付ける深層学習アルゴリズムであり、特徴的なカテゴリを形成する画素の集まりを認識するために使用される。画像内を画素レベルで複数の領域に分けることができるため、境界ボックス内に目標のオブジェクトが収まらなければならない物体検出法に比べて不規則な形状の対象物の検出を可能としている [5].

本研究では、図 1 のような地元メーカーが製造する工業製品に縦方向に発生するヒビ状の傷 (欠陥) の検出に SS を適用し、その有効性を検証する。なお、現状の製造工程の制約から画像は作業者がカメラを手に持ちマニュアル撮影した写真であり、製品上部にはブルーシートが含まれ、さらに製品部分には傷に似た手書きの文字などが含まれている。メーカーから良品 31 枚、不良品 72 枚の計 103 枚の工業製品の画像を提供いただき、不良品画像のうち 37 枚に対して欠陥部分のピクセルラベリングを行い、学習用に用いた。具体的には、まず、この 37 枚の不良品画像に対して MATLAB 上で利用できるアノテーションアプリケーション「イメージラベラー」を用いて背景の部分 (黄色)、製品の部分 (青色)、および傷の部分 (水色) の三つのラベル付けを行い、学習用 gTruth データを作成し、これをデータセット A とした。このデータセット A に水平反転画像を加え、計 74 枚とした gTruth データをデータセット B とした。データセット B の各画像の背景に対して、別に用意した 75 枚の画像からランダムに抽出した 5 枚の画像を貼り付けることで  $74 \times 5 = 370$  枚の画像が得られ、

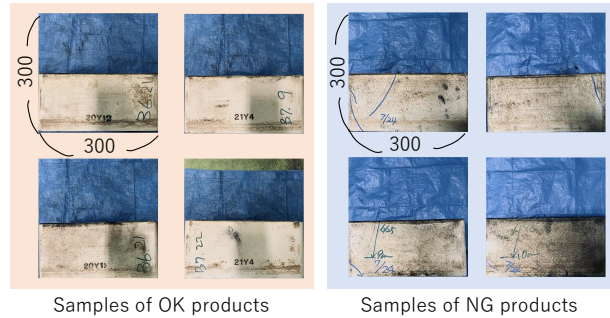


Figure 1 Image samples used for training a U-Net model. Crack-like defects are seen in images of NG products.

これと元のデータセット B を加えて計 444 枚としたものをデータセット C とした。さらに、データセット C の中で、目視でより明確に傷と確認できるものだけを再選択し、傷をしっかりと覆うようにラベリングをやり直したものをデータセット D とした。

つぎに、SS 学習用モデルとして図 2 のようなエンコーダとデコーダそれぞれの深さが 2 の U-Net 層 [6] を設計した。入力画像がエンコーダによりダウンサンプリングされる回数、デコーダによりアップサンプリングされる回数が深さとなる。U-Net はエンコーダのサブネットワークと、それに対応するデコーダのサブネットワークで構成されており、エンコーダの特徴マップをデコーダの特徴マップに連結させるスキップ構造を持つ。それぞれのデータセット (gTruth データ) をもとに Adam (Adaptive moment estimation: 適応モーメント推定) オプティマイザ [7] で学習したモデルは U-Net A, U-Net B, U-Net C, U-Net D として保存した。これら 4 つのモデルに良品と不良品画像それぞれ 31 枚、35 枚のテスト用データを与え、画像内で傷として判定された領域の面積 (領域を構成するピクセル数) から分類精度を評価した。今回は、しきい値を 140 に設定し、欠陥部分の面積が 140 以上の場合は不良品として判定している。

なお、各 U-Net からの出力は入力画像と同じ解像度を持つマップであり、各ピクセルは背景、製品、傷といった分類結果に応じてそれぞれ 1, 2 あるいは 3 の値を持つ。このため、傷と推定されたピクセルの値を 3 から 255 にし、それ以外のピクセルの値を 0 とすることで二値化画像が得られるため、イメージ領域のプロパティの計測が可能な regionprops 関数をコールすることで 8 連結により傷の特長を持つ領域を抽出することができる。

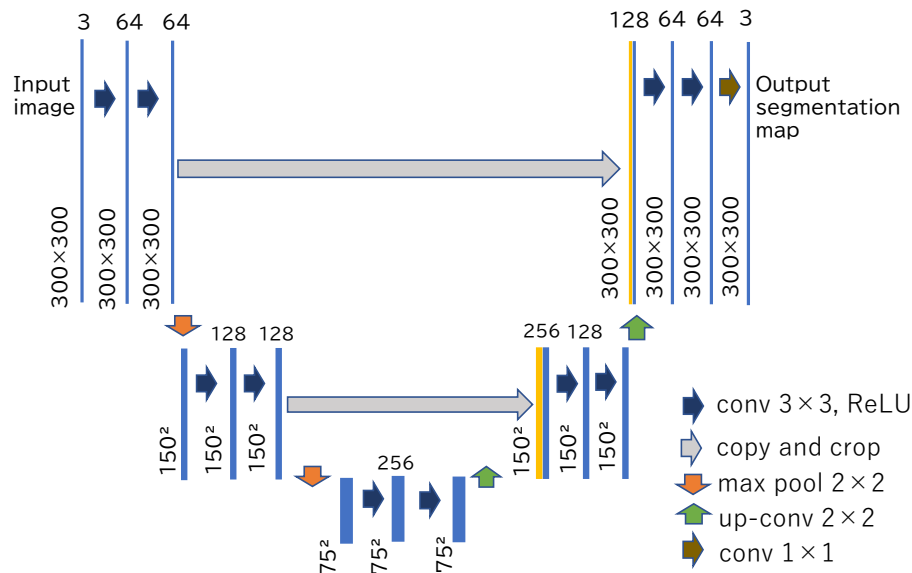


Figure 2 The authors' originally constructed U-Net type model for the detection of crack-like defects.

### 3 実験結果

4つの学習モデルのうち3カテゴリ全てを出力できていたU-Net CとU-Net Dの分類結果を比較したところ、U-Net Dでは良品、不良品とも96%を超える精度で分類できていたことが確認された。図3にはU-Net Dにテストデータを与えて出力された35枚の不良品画像のうちの6枚を例として示している。黄色が背景、青色が製品、水色がひび上の欠陥を表しており、3つのカテゴリとも良好に出力できていることが確認された。以上の結果から、ランダム画像による背景変更、学習画像枚数の拡張、学習エポック数の調整などが重要だと分かった。また、YOLOv2では難しかった傷の面積や長さのしきい値に基づく定量的な確認と欠陥検出も可能になった。

今後は、欠陥部分の長さや面積による定量的な評価に基づく分類機能を必要とする他の工業製品への適用を試みたい。また、今回のモデル設計、学習作業、検証実験などはMATLAB上で行ったきたが、分類結果に基づき良品ロットあるいは不良品ロットにプレイングを行うロボットがPython上で動作している場合など、MATLABからエクスポートさせたONNXモデル[8]をPython上で再利用したいというニーズがあり、このような異なる開発環境での動作の同等性(相互運用性)についても評価していきたい。

### References

[1] 阿部凌真, 有馬滉宜, 清水竜樹, 永田寅臣, 加藤博久, 渡辺桂吾, “Grad-CAMによる欠陥領域の可視化性能を向上させる画像拡張法とCNNによる分類時のスコアの影響”, 第28回画像センシングシンポジウム講演論文集, 4 pages, 2022.

[2] 松山弘樹, 永田寅臣, 加藤博久, 渡辺桂吾, “金属の火花試験の自動評価のための3D CNNの適用”, 第28回画像センシングシンポジウム講演論文集, 4 pages, 2022.

[3] J. Redmon, A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *Procs. of the IEEE Conference*

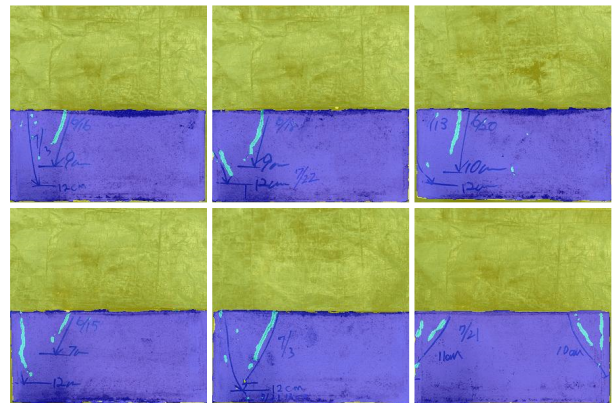


Figure 3 Image samples of an industrial product, in which crack-like defects are visually included.

*on Computer Vision and Pattern Recognition*, pp. 6517–6525, 2017.

[4] LC. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”, *Procs. of European Conference on Computer Vision 2018 (ECCV2018)*, pp. 833–851, 2018.

[5] <https://jp.mathworks.com/>

[6] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Vol. 9351, pp. 234–241, 2015.

[7] D.P. Kingma, J.L. Ba, “Adam: A method for stochastic optimization,” *Procs. of the 3rd International Conference on Learning Representations (ICLR2015)*, pp. 1–15, 2015.

[8] <https://onnx.ai/>