

Python上で動作するピックアンドブレースロボットののための ONNX Runtime モデルを用いた欠陥検出機能

坂田 真吾† 永田 寅臣† 阿部 凌真† 池田 毅† 加藤 博久† 渡辺 桂吾‡

† 山口東京理科大学 ‡ 岡山大学

E-mail: nagata@rs.socu.ac.jp

1 はじめに

筆者らはこれまでに、工業製品や工業材料の不良品検出のための CNN, SVM, CAE, FCN, YOLO などの深層学習モデルを効率的に設計、訓練、評価できるアプリケーションを MATLAB の AppDesigner 上で開発してきた [1, 2]。最近ではそのような深層学習モデルを異なるフレームワーク間で相互運用したいといったニーズが高まっている。ONNX は相互運用可能な機械学習モデルを表現するために構築されたオープンデータ形式である。ONNX 形式に変換することにより、異なるフレームワーク間での機械学習モデルの共有を可能にしている [3]。今回、MATLAB フレームワーク上で構築したある工業製品の欠陥検出用 CNN モデルを Python 環境下で動作しているピックアンドブレースロボットで活用したいというニーズがあった。このため、MATLAB 上で構築している CNN モデルについては ONNX (Open Neural Network eXchange) モデルへエクスポート後に、Python 側ではそれを Runtime モデルとしてインポートすることで、また、訓練やテストに用いる画像については各ピクセルが持つ数値レベルでの完全な互換性が保たれるような解像度変換を行うことで、異なるフレームワーク間でも同じ分類性能を維持しながら活用できるようなシステムについて検討したので報告する。また、簡易操作で利用できるように開発したユーザインタフェースとその評価結果についても紹介する [4, 5]。

2 MATLAB 環境での CNN モデルの構築

2.1 VGG19の転移学習ベースのCNNモデルの設計

これまでに、工業製品、工業材料、培養細胞、金属の火花試験などの写真の分類問題へ適用するための様々な CNN モデルを MATLAB 上で構築してきた中では、VGG19 の転移学習ベースのモデルが常に高い分類精度を発揮できていたために、今回も同様の転移学習による設計方法で CNN モデルを構築することとした。ある工業製品の製造メーカーから提供のあった JPEG 形式の良品と不良品のオリジナル画像はそれぞれ 156 枚と

196 枚であったため、これらを CNN の訓練に用いた。また、簡単な画像処理で左右反転させた 156 枚と 196 枚を学習後の CNN のテスト用、すなわち汎化性能の評価に用いた。

図 1 には、OK あるいは NG の 2 クラス分類のための VGG19 の転移学習ベースの CNN モデルのネットワーク構造を示している。本研究で使用したオリジナル画像の解像度は $2590 \times 1942 \times 3$ であったが、転移学習時には VGG19 の入力層に合わせて $244 \times 244 \times 3$ にダウンサイジングする必要がある。MATLAB が提供する画像の解像度変換の関数は `imresize()` であり、‘NEAREST’, ‘BILINEAR’, ‘BICUBIC’ などの内挿法を利用できる。‘NEAREST’ は最近傍内挿法であり、出力ピクセルには対象となる点のピクセルの値がそのまま代入される。‘BILINEAR’ と ‘BICUBIC’ はそれぞれ、双一次内挿法と双三次内挿法であり、最近傍の 2×2 あるいは 4×4 のピクセルの重み付き平均値が代入される。今回は、オリジナル画像が持つピクセル値をそのまま適用するために ‘NEAREST’ をパラメータとして設定し、VGG19 の入力層に合わせてダウンサイジングを行った。

図 2 には可視化結果の例を示しており、それぞれ、Nearest と Bilinear, Nearest と Bicubic, Bilinear と Bicubic のように異なる補間方法でダウンサイジングを行った画像のピクセル値の差分を示している。このあとの実験では Nearest によって縮小された画像を CNN の訓練、汎化性能のテスト、Python 環境下との互換性の評価に使用している。

2.2 ONNX (Open Neural Network Exchange)

ONNX は 2017 年ころから Microsoft と Facebook が共同で開発を開始した統一的なフォーマットである。ONNX が提案される以前は、ニューラルネットワークモデルが様々なフレームワーク上で学習され、各フレームワークに特化したモデルとなってしまう、他のフレームワークで利用することが容易ではなかった。ONNX が提案、開発されることによって、どのフレームワークで学習が行われていたのかを問わずに、学習させたモデルを ONNX 形式に変換することで異なるフレームワーク上で相互運用することが可能となった [3]。

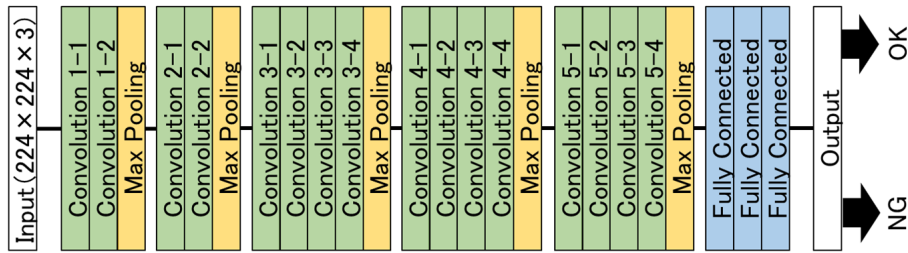


図 1 Network structure of a CNN model for binary classification, i.e., OK or NG, which is transferred from original VGG19 model.

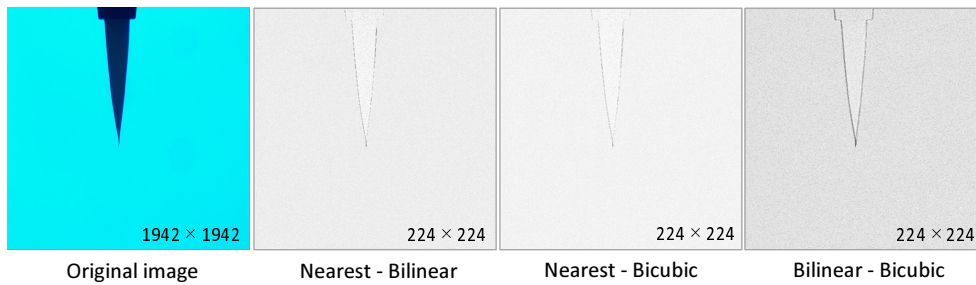


図 2 Comparison examples among three types of image interpolation methods.

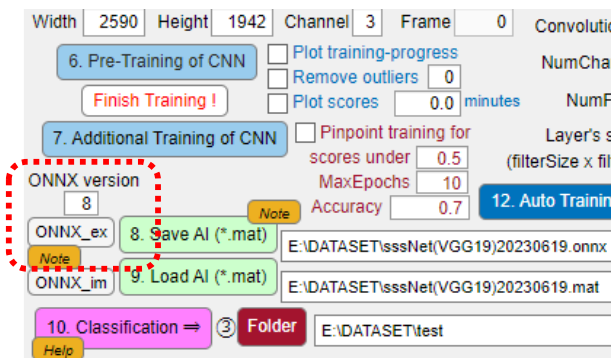


図 3 Part of ONNX model export function included in the developed MATLAB application.

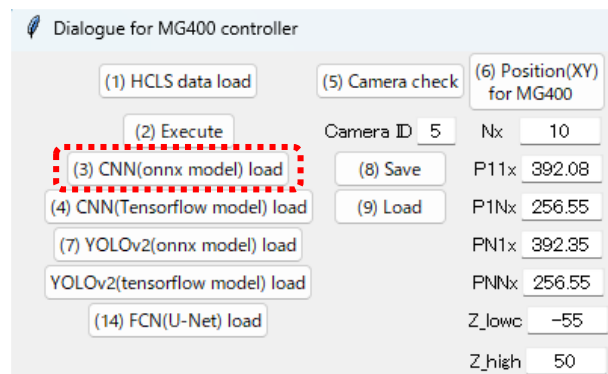


図 4 Part of ONNX model import function included in the developed Python application for a picking robot control.

2.3 ONNX モデルへのエクスポート

MATLAB 上で構築した VGG19 の転移学習ベースの CNN モデルは、既に開発している図 3 に示すような MATLAB アプリ上の ONNX Export ボタンにより `exportONNXNetwork()` 関数をコールすることで ONNX モデルに変換できるようにしている。サポートされている ONNX 演算子セットは 6~14 であり、今回は規定値である 8 に設定し、実行することで Python 側にインポート可能な ONNX モデルのファイル (拡張子: onnx) を生成した。

3 Python 環境下での ONNX モデルを用いた分類性能の評価

Python 上で開発中の小型ロボット MG400 制御用のアプリケーションでは、図 4 の ONNX モデルのインポート機能により ONNX モデルをインポートすることができるため、ONNX Runtime を使った画像分類が可能となる。これら異なるフレームワーク上で同一の分類スコアを得るためには、MATLAB 上で実行したオリジナル画像のダウンサイジングと、ピクセル値のレベル

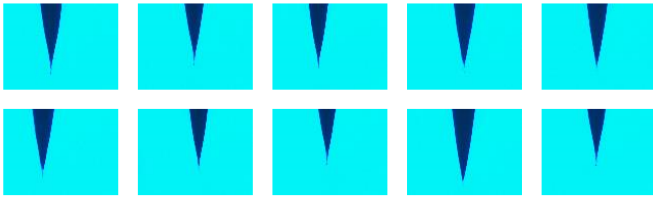


図 5 Examples of enlarged tip images of non-defective products.

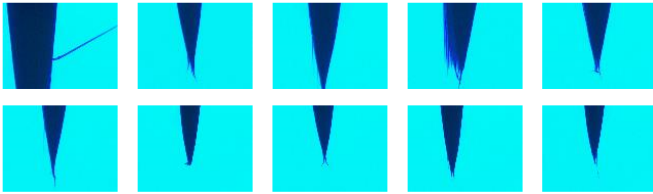


図 6 Examples of enlarged tip images of defective products.

で完全に同値となるようなダウンサイジングを Python 上でも行う必要がある。

Python 上では当初、コンピュータビジョンライブラリの `cv2.resize()` 関数により解像度の変換ができたために内挿法に 'NEAREST' を指定し、テスト画像のサイズを $244 \times 244 \times 3$ に縮小後にインポートした ONNX モデルの入力層に与えていたが、MATLAB 上で得られていた分類スコアとは大きく異なる値が出力されていた。このため、MATLAB の `imresize()` 関数で $244 \times 244 \times 3$ にサイズ変更した画像と、Python 上で `cv2.resize()` 関数により同様に処理した画像を比較したところ、多くのピクセルで値の違いが見受けられた。

さて、Python 上では画像処理ライブラリの Pillow が提供する `resize()` 関数も利用できたため、'NEAREST' を指定した上で複数のテスト画像に対して同様のダウンサイジングを行い、ONNX モデルでの分類実験を行った。図 5 と図 6 には、評価実験に用いた良品 10 枚と不良品 10 枚の先端画像をそれぞれ示す。また、表 1 と表 2 には、MATLAB 上での CNN モデルと Python 上の ONNX モデルがそれぞれの画像を評価したときのスコアを示す。これらの結果から、MATLAB 上での分類結果と同値の分類スコアを Python 上でも得られることが確認できた。

つぎに、'NEAREST' と同様に 'BILINEAR' と 'BICUBIC' についても分類実験を行った。まず、それぞれのフレームワークで 'BILINEAR', 'BICUBIC' を指定し、ダウンサイジングを行った画像を比較すると多くのピクセルで異なる値が確認された。図 7 (左) には、MATLAB 上と Python 上で 'BILINEAR' を指定しダウンサイジングした画像の差異を求め、差異のあるピクセルを白、差異のない同値のピクセルを黒で表示した図を示している。また、図 7 (右) は 'BICUBIC' のときの結果である。このように、'BILINEAR', 'BICUBIC' を指定しダウンサイジングすると一部のピクセルに差異が生じ、結果的に CNN が出力する分類スコアが異なってしまうため、今回の実験においては適切なダウンサイジング法ではないことが確認された。

表 1 Classification accuracies of 224×224 downsized images of non-defective materials.

	MATLAB	Pillow (Python)	CV2 (Python)
Trial 1	0.902	0.902	0.986
Trial 2	0.953	0.953	0.976
Trial 3	0.987	0.987	0.983
Trial 4	0.839	0.839	0.699
Trial 5	0.949	0.949	0.812
Trial 6	0.708	0.708	0.990
Trial 7	0.867	0.867	0.814
Trial 8	0.969	0.969	0.953
Trial 9	0.854	0.854	0.981
Trial 10	0.992	0.992	0.972
Average	0.902	0.902	0.916

表 2 Classification accuracies of 224×224 downsized images of defective materials.

	MATLAB	Pillow (Python)	CV2 (Python)
Trial 1	1.000	1.000	0.999
Trial 2	0.989	0.989	0.999
Trial 3	0.991	0.991	0.391
Trial 4	0.999	0.999	0.999
Trial 5	0.972	0.972	0.999
Trial 6	1.000	1.000	0.999
Trial 7	0.999	0.999	0.993
Trial 8	0.959	0.959	0.033
Trial 9	0.996	0.996	0.636
Trial 10	0.999	0.999	0.757
Average	0.990	0.990	0.780

セルを白、差異のない同値のピクセルを黒で表示した図を示している。また、図 7 (右) は 'BICUBIC' のときの結果である。このように、'BILINEAR', 'BICUBIC' を指定しダウンサイジングすると一部のピクセルに差異が生じ、結果的に CNN が出力する分類スコアが異なってしまうため、今回の実験においては適切なダウンサイジング法ではないことが確認された。

4 ONNX モデルを用いたロボットによるペグインホールタスク実験

前章で確認したように、Python の画像処理ライブラリ Pillow が提供する画像リサイズ関数の補間方法に 'NEAREST' を指定し画像をダウンサイジングすることで、MATLAB と Python という 2 つのフレームワー

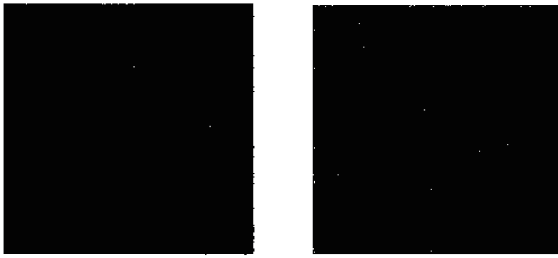


図 7 Comparison of Bilinear and Bicubic interpolations on different types of frameworks as MATLAB and Python.

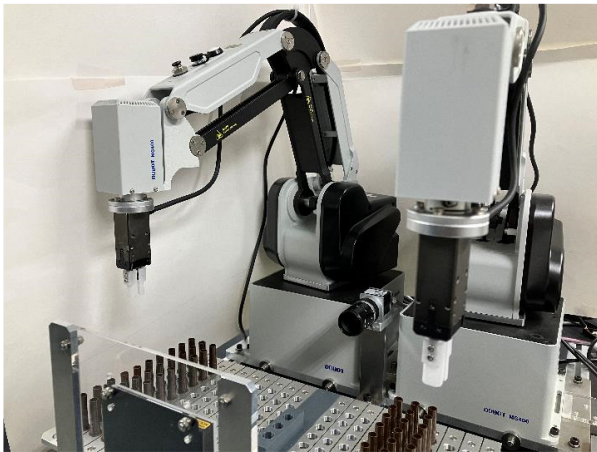


図 8 Overview of the proposed cooperative robots system running on Python for a peg-in-hole task while detecting undesirable defects.

ク間でピクセル値のレベルで CNN モデルの相互運用が可能となった。ここでは、Python 上で動作している 2 台の小型産業用ロボット MG400 の制御アプリケーション内に欠陥検出用 ONNX モデルをインポートし、一つのカメラを共用しながら協調的に良品と不良品を自動仕分けするペグインホールタスクに適用した。図 8 には、ロボットシステムによる複数ワークのペグインホールタスクの実行風景を示す。実験の結果、ロボットはワークを一つずつピックアップし、カメラ前方へ移動させて 120 度ずつ回転させながら 3 方向からの写真を撮影し、ONNX モデルで評価し、3 枚とも良品と判定された場合に OK 領域へ、一枚でも不良品と判定された場合は NG 領域へプレーシングする動作を終了まで連続的に実行できることを確認した。

5 おわりに

本研究では、MATLAB 上で構築した工業製品の欠陥検出用 CNN モデルを Python 環境下で動作してい

る自動ピックアップロボットで利用するための検証実験を行った。VGG19 の転移学習により構築したシリーズ型の欠陥検出用 CNN モデルについては ONNX (Open Neural Network eXchange) モデルへエクスポート後に、Python 側ではそれを Runtime モデルとしてインポートすることで、また、訓練やテストに用いる画像については各ピクセルが持つ数値レベルでの完全な互換性が保たれるように画像処理ライブラリ Pillow が提供するリサイズ関数を用いてダウンサイジングを行うことで、異なるフレームワークにおいても同じ分類性能を維持しながら活用できるシステムを実現できた。簡易操作で利用できるように MATLAB 上および Python 上で開発したユーザインタフェースの有用性についても確認することができた。

参考文献

- [1] F. Nagata, K. Tokuno, H. Ochi, A. Otsuka, T. Ikeda, K. Watanabe and M. K. Habib, “A Design and Training Application for Deep Convolutional Neural Networks and Support Vector Machines Developed on MATLAB,” *Procs. of the 6th International Conference on Robot Intelligence Technology and Applications (RiTA2018), Lecture Notes in Mechanical Engineering (LNME)*, Springer, pp. 27–33, 2018.
- [2] F. Nagata, M.K. Habib, K. Watanabe, “Transfer Learning-Based and Originally-Designed CNNs for Robotic Pick and Place Operation,” *International Journal of Mechatronics and Automation*, Inderscience Publishers, Vol. 8, No. 3, pp. 142–150, 2021.
- [3] <https://onnx.ai/>
Accessed 25 November 2023.
- [4] 坂田真吾, 阿部凌真, 永田寅臣, “ONNX の Runtime モデルを用いた自動ピックアップロボットのための欠陥検出機能”, 第 31 回 インテリジェントシステムシンポジウム FAN2023 講演論文集, Th-A3-2(1-2), 九州大学椎木講堂, 2023.
- [5] S. Sakata, F. Nagata, R. Abe, K. Watanabe, “Defect Detection System for an Automatic Picking Robot Using an ONNX Runtime Model,” *Procs. of The 29th International Symposium on ARTIFICIAL LIFE AND ROBOTICS (AROB 29th 2024)*, pp. 943-946, 2024.