AI 技術を活用した検査工程の省力化・効率化（第６報）

－学習済みモデルを利用した欠陥画像の分類－

渡辺博己*、生駒晃大*、松原早苗*

A study on artificial intelligence for labor savings and efficiency improvements of inspection process（Ⅵ）

－Defect images classification using pre-trained models－

WATANABE Hiroki*, IKOMA Akihiro*, MATSUBARA Sanae*

限られたデータ数でも予測精度の高い深層学習モデルを生成することが可能であることから、転移学習やファインチューニングを用いた学習済みモデルの利用が注目されている。本研究では、欠陥画像データセットに対する学習済みモデルの分類性能を検証し、分類性能が高い学習済みモデルを適用した欠陥画像分類モデルを構築した。実験では、学習済みモデル適用前後の分類性能を比較し、適用後のモデルの有効性を確認した。

１．はじめに

近年、深層学習により生成されたモデル（以下、深層学習モデル）の予測精度が高いことから、深層学習モデルが社会生活の様々な場面で使用されるようになりつつある。深層学習モデルが高い予測精度を持つ理由の一つとして、予測を行う上で非常に重要な特徴量を抽出する仕組みを深層学習モデルが獲得していることが挙げられる。しかし、このようなモデルを生成するためには、大量のデータが必要であることが知られており、データ収集が予測精度の高い深層学習モデル生成の課題となっている。

この課題を解決する手法として、転移学習やファインチューニングと呼ばれる手法が有効であると言われている。これらの手法は、大量のデータで学習されたモデル（以下、学習済みモデル）を特徴量抽出器として再利用することで、限られたデータ数でも高い予測精度の獲得を可能にしている。

本研究では、複数の学習済みモデルに対して、転移学習とファインチューニングによる欠陥画像の分類性能を検証した。また、従来の欠陥画像分類モデル [1]（以下、従来モデル）に学習済みモデルを適用したモデルを生成し、画像分類性能を検証した。本稿では、これらの内容について報告する。

２．学習済みモデルによる欠陥画像分類

２．１　転移学習とファインチューニング

転移学習やファインチューニングは、学習済みモデルを利用する際に使用される手法である。転移学習は、学習済みモデルの重みデータを変更せずに、特徴量抽出器として利用する手法である。また、ファインチューニングは、学習済みモデルの重みデータを初期値として、重みデータの一部、または全部を再学習して、特徴量抽出器として利用する手法である。図１に、転移学習とファインチューニングの概念を示す。なお、本研究では、

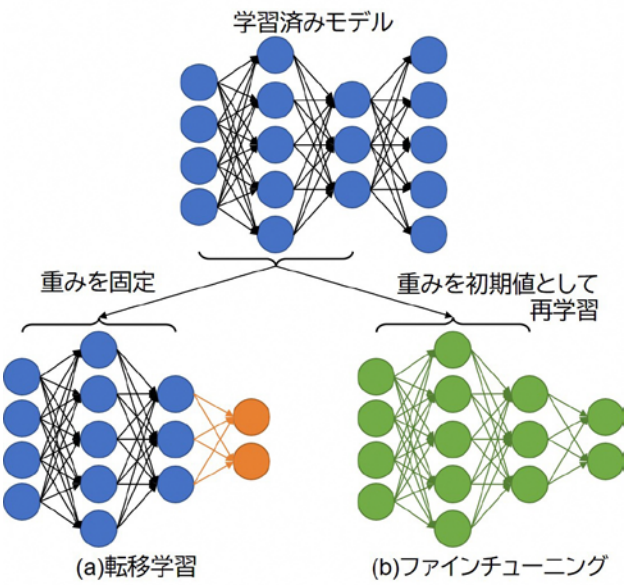

図１　転移学習とファインチューニング

ImageNet データセット [2]により重みデータを学習したモデルを利用した。

２．２　学習済みモデルの比較実験

実験では、学習済みモデルとして VGG[3]、ResNet[4]、EfficientNet[5]の三種類を利用し、各学習済みモデルに対して、転移学習、ファインチューニング、及び重みデータを初期値として利用しない一般的な教師あり学習（以下、通常学習）を行った。表１に、本実験で用いた欠陥画像データセット [1]の各クラスのデータ数を示す。

表２に結果を示す。結果は、検証用データの各クラスに対する正解数の和を、検証用データの総和で除した、全クラスに対する正解率である。また、VGG は、16、19 層モデル、ResNet は、50、101、152 層モデル、EfficientNet は、B0 から B7 までの各モデルについて、転移学習、ファインチューニング、及び通常学習を行い、それぞれの手法における正解率を求めた。なお、参考として、従来モデルにおける正解率も示す。

結果より、ファインチューニング、転移学習、通常学

＊ 情報技術部

表１　欠陥画像データセットの各クラスのデータ数

欠陥A	602 100	欠陥F	367 100	欠陥K	352 100	正常 （異物）	2,344 100
欠陥B	336 100	欠陥G	133 100	欠陥L	247 100	正常 （キズ）	3,133 100
欠陥C	231 100	欠陥H	192 100	欠陥M	407 100	正常 （汚れ）	2,374 100
欠陥D	278 100	欠陥I	281 100	欠陥N	319 100	正常	3,346 100
欠陥E	299 100	欠陥J	571 100	欠陥O	104 100	計	15,916 1,900

※上段:学習用データ数、下段:検証用データ数

表２　学習済みモデルの正解率

学習済みモデル		転移学習	ファイン チューニング	通常学習
VGG	16	0.7163	0.8663	0.7100
	19	0.7016	0.8621	0.6905
	平均	0.7090	0.8642	0.7003
ResNet	50	0.7389	0.7705	0.6526
	101	0.7400	0.8374	0.6432
	152	0.7311	0.8205	0.6742
	平均	0.7367	0.8095	0.6567
EfficientNet	B0	0.8037	0.8863	0.4732
	B1	0.8111	0.8853	0.4826
	B2	0.8263	0.8842	0.4747
	B3	0.8111	0.8879	0.5274
	B4	0.8116	0.8837	0.5442
	B5	0.8100	0.8821	0.5416
	B6	0.8089	0.8858	0.5663
	B7	0.8116	0.8921	0.5232
	平均	0.8118	0.8859	0.5167
従来モデル				0.9221

習の順に平均正解率が高くなった。そのため、欠陥画像データセットに対して学習済みモデルを利用する場合は、ファインチューニングを用いることが有効であることが分かった。また、従来モデルの正解率には及ばないものの、学習済みモデルとしては EfficientNet の平均正解率が最も高くなった。

３．学習済みモデルの適用

３．１　欠陥画像分類モデルへの適用

従来モデルでは、256×256 のサイズで入力された画像を 160×160 に縮小した画像と、160×160 で画像中心を切り取った画像の 2 つの画像を入力データとして使用する。それぞれの入力データからは、12 個の畳み込み層と 4 個のプーリング層を持つ同一構成の 2 つのネットワークにより、それぞれ特徴マップが生成される。従来モデルは、2 つの特徴マップを統合し、畳み込みを行った後、GAP により特徴量を抽出し、全結合層を経て各クラスに対する確率を計算する。

従来モデルにおける特徴マップ生成ネットワークは、VGG をベースに、スキップコネクションや SE ブロック[6]等を加えたネットワーク構成であるが、２．２の実験により、ファインチューニングによる EfficientNet の正解率が最も高かったことから、本研究では、特徴マップ生成ネットワークに EfficientNet を適用したモデル（以下、適用モデル）を構築した。図２に、適用モデルの構成を示す。

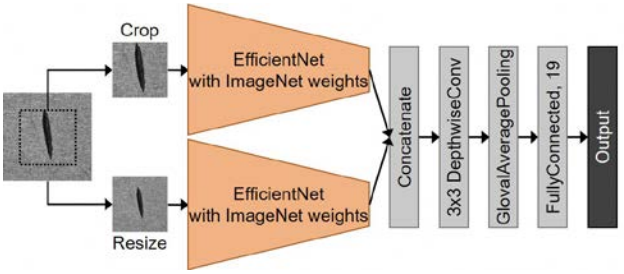

図２　EfficientNet 適用後の欠陥画像分類モデルの構成

表３　学習済みモデル適用前後の正解率

	従来モデル	適用モデル (EfficientNet-B0)	適用モデル (EfficientNet-B4)
欠陥クラス	0.9280	0.9300	0.9420
正常クラス	0.9000	0.8900	0.8875
全クラス	0.9221	0.9216	0.9305

３．２　適用モデルによる欠陥画像分類実験

実験では、２．２と同様、従来モデルと適用モデルについて、学習用データをファインチューニングにより学習し、検証用データに対する全クラスの正解率と、欠陥クラス、正常クラスの正解率を求めた。なお、適用モデルについては、EfficientNet-B0 と EfficientNet-B4 を利用した場合の 2 つの実験を行った。表３に結果を示す。

結果より、EfficientNet-B0 による適用モデルでは、僅かに従来モデルに及ばなかったものの、EfficientNet-B4 による適用モデルでは、正解率が 0.84%向上した。ただし、欠陥、正常クラス別では、欠陥クラスで 1.40%上がったのに対して、正常クラスで 1.25%下がったため、正常クラスに対する正解率の向上が課題である。

４．まとめ

本研究では、学習済みモデルの欠陥画像データセットに対する分類性能を検証した。また、分類性能が高かった EfficientNet を利用して、欠陥画像分類モデルを改良した。その結果、従来モデルの正解率を 0.84%向上することができた。

今後も、深層学習におけるモデル構成技術の習得を継続し、さらなる分類性能の向上を目指す。

【謝　　辞】

本研究を遂行するにあたり、欠陥画像データセットをご提供いただきました株式会社前田精工の皆様に深く感謝の意を表します。

【参考文献】

1)　渡辺ら，岐阜県産業技術総合センター研究報告，No.1, pp.81-84, 2020
2)　O. Russakovsky, et al., arXiv:1409.0575, 2014
3)　K. Simonyan, et al., arXiv:1409.1556, 2014
4)　K. He, et al., arXiv:1512.03385, 2015
5)　M. Tan, et al., arXiv:1905.11946, 2019
6)　J. Hu, et al., arXiv:1709.01507, 2017