報　文

徳島県立工業技術センター研究報告 vol.31 2022

少数サンプルにおけるAIの精度改善に関する研究

Research on Improvement of AI Accuracy in Few Samples

平岡　忠志*
HIRAOKA Tadashi

抄　録

教師あり画像分類問題について，少数サンプルにおいても高精度な人工知能（AI）のモデルを作成するため，Computer Graphics（CG）画像を用いた転移学習について検証した．金属研削面のキズの有無を判断する2クラス分類問題において，各クラスの学習用画像枚数を1から128まで変化させたとき，モデルの初期パラメータを乱数とした通常の学習及びImageNetを用いた転移学習のモデルと比較して，CG画像を用いた転移学習のモデルがより高精度だった．CG 画像を用いた転移学習により少数サンプルにおけるモデルの精度改善が可能であることを示した．

1　はじめに

製造業では人手不足の問題が深刻化しており，製品の品質を確保する要である外観検査においても検査員の不足は深刻である．また，各検査員による判断基準の曖昧さや不良品の見逃しという課題もある．このような課題に対応するために従来から画像処理を利用した外観検査の自動化が進められたが，期待する精度が出ないという問題があった．

近年，人工知能（AI）の1つの技術である深層学習の活用により外観検査の高精度化が図られているが，多数のサンプル収集及びラベル付けが必要であるという問題がある．これは，画像処理においては人が特徴抽出方法を設計するのに対し，深層学習においては機械が多数の画像から特徴抽出方法を自動的に設計するため，深層学習を利用するには必然的に多数の画像が必要となるからである．

この問題に対し，多数の画像により特徴抽出方法が学習されたモデルのパラメータを転用し，別の少数の画像分類タスクに適用する転移学習が有効であることが示された[1)]．しかし，転移元と転移先のクラスの分布が大きく異なる場合は良い結果が期待できない．例えば，転移元に動物等の画像で構成されたデータセットを利用し，転移先に金属研削面のキズの有無を持つ画像で構成されたデータセットを利用した場合は低精度となる．

この問題を解決するもう一つの方法として，Computer Graphics（CG）技術により疑似画像を生成し転移学習する方法が考えられるが，この場合は現実世界から得られた画像とCGから得られた画像とのギャップが問題となる．Tremblayらは自動車認識の問題において光源及び物体のテクスチャをランダムに変化させるDomain Randomizationを用いてモデルを高精度化させた[2)]．

本研究では解決したい問題を金属研削面のキズの有無を判断する2クラス分類問題とし，少数サンプルにおいても高精度なモデルを作成できる Domain Randomization によって生成された CG 画像を用いた転移学習について検証した．

2　方法

2・1　深層学習の問題点

深層学習は，積和からなる線形関数と正規化線形関数等の非線形関数を幾層にも重ねたモデルを用いることで，多数の画像を基にして局所から大域までの段階的な特徴抽出方法を自動的に設計する点に特徴があり，深層学習によって高精度なモデルが作成可能である．しかし，多数のサンプルが必要とされるため，そのままでは外観検査の自動化への適用は困難である．以下に多数のサンプルを用いた場合の通常のモデルの作成方法と問題点を示した．

（1）画像の表現方法

画像は空間上の点として表現可能である．例えば，224×224画素の1枚の画像は50,176次元空間上の1点と表現される．図1は可視化のため，50,176次元

＊ 機械技術担当

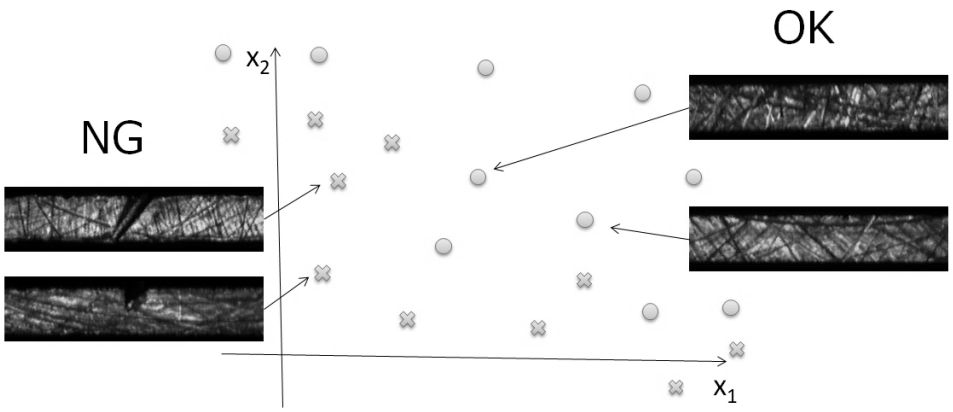
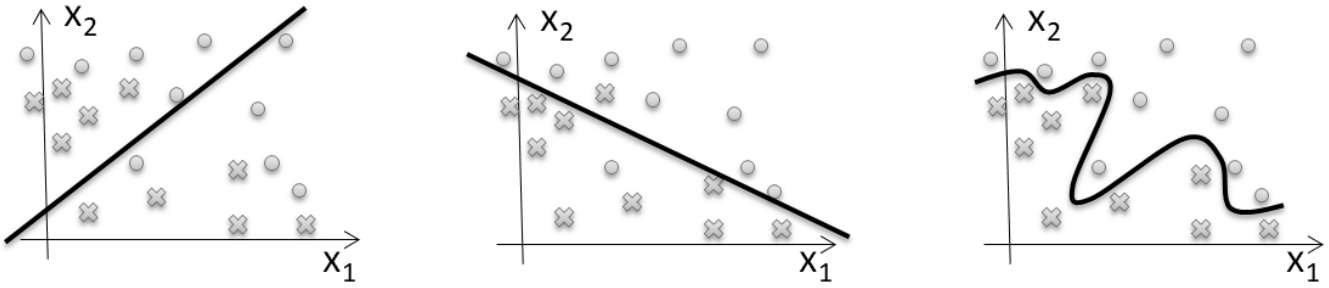
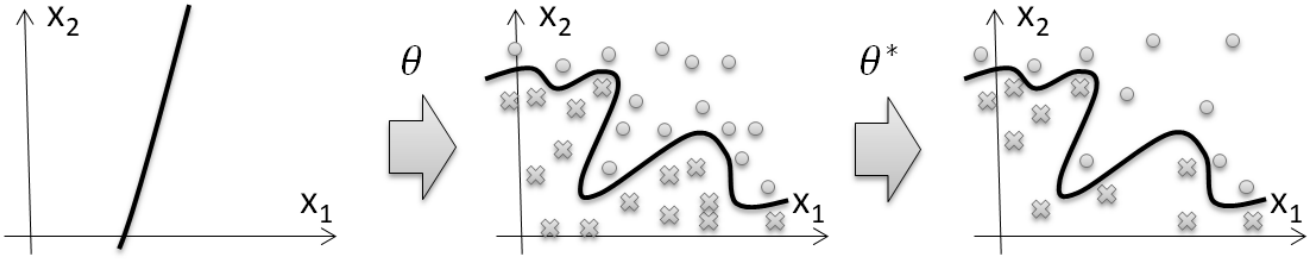


図１　画像の表現方法



（a）精度 50%　　（b）精度 90%　　（c）精度 100%

図２　パラメータ θ の変更によるモデルの分類精度の違い



（a）初期化　　（b）学習　　（c）評価

図３　通常の学習と評価の流れ（学習に多数のサンプルが必要）

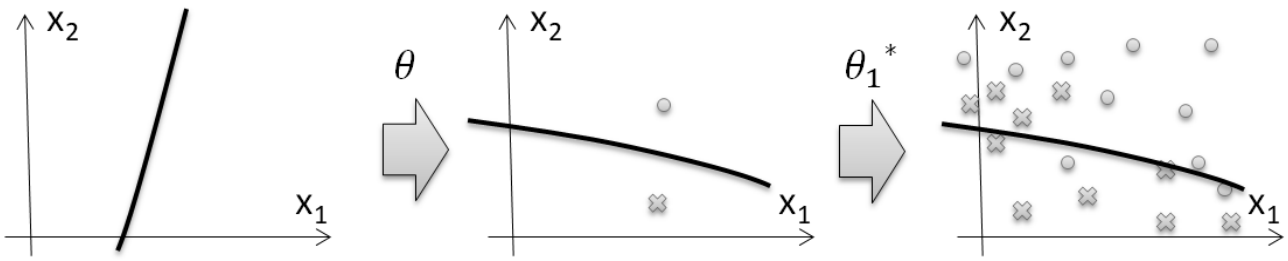
空間から x_1 と x_2 の 2 次元を選択し, OK 画像 10 枚, NG 画像 10 枚を描画した図である．つまり，50,176 画素から 2 画素を選択したものである．

（２）モデルによる空間分割

深層学習の目的は，この 20 点を参考にして空間を OK と NG の 2 つの空間に分割するようなモデルを作成することである．ここでのモデルは関数または曲線とも言い換えることができる．図２のようにモデルのパラメータ θ を変更することで様々な表現が可能である．深層学習でよく利用されるモデルのパラメータ数は 1 千万以上あり，パラメータ θ は目的関数と多数のサンプルを利用して確率的勾配降下法により推定される．パラメータ θ を持つ図 2(a)のモデルは 20 点のうち 10 点を正しく分類できる．この場合，精度は 50%と低精度である．ここで，精度は正しく分類できた割合で定義される．改良したパラメータ θ を持つ図 2 (b)のモデルは 20 点のうち 18 点を正しく分類できるので，精度は 90%である．図 2(c)のモデルの精度は 100%であり，高精度である．

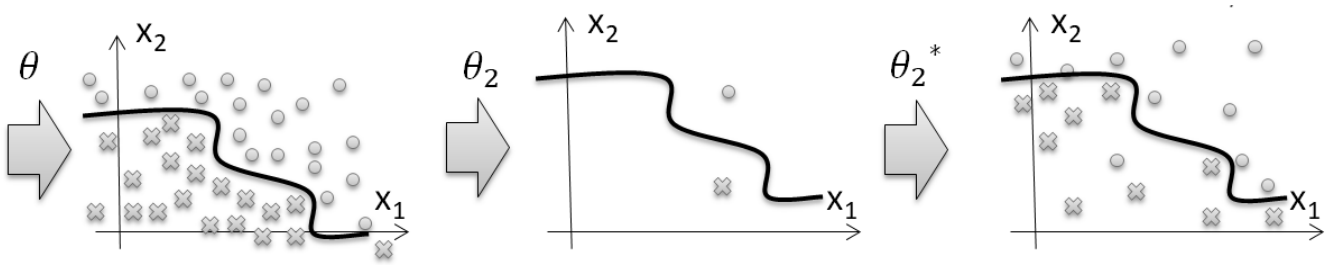
（３）通常の学習と評価の流れ

図 3 に図 2(c)のような高精度なモデルを作成する手順を示した．まず，パラメータ θ を乱数で初期化する（図 3(a)）．次に，約 100 万枚の学習用画像を用いてパラメータ θ を最適化し θ^*を得る（図 3(b)）．最後に約 5 万枚の評価用画像を用いてパラメータ θ^*を持つモデルを評価する（図 3(c)）．学習/評価用画像に



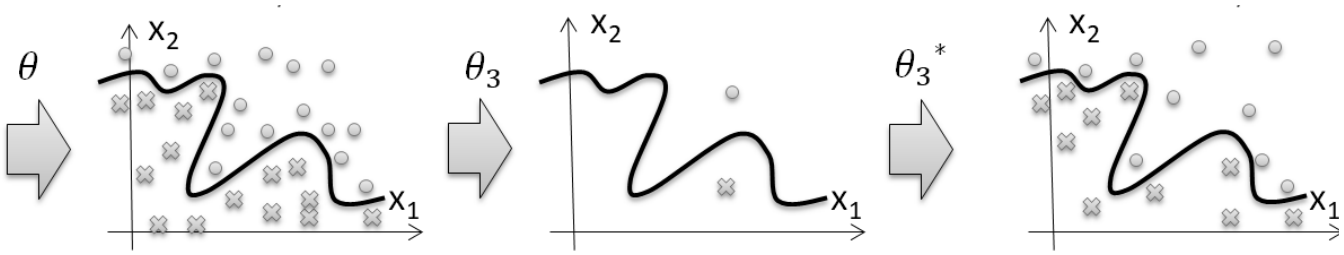
（a）初期化　　（b）学習　　（c）評価

図４　**Scratch**



（a）事前学習　　（b）学習　　（c）評価

図５　**ImageNet Pretrain**



（a）事前学習　　（b）学習　　（c）評価

図６　**CG Pretrain**

は現実に適用したい画像を用意する．このように通常の学習では多数の学習用画像が必要なので，この方法では外観検査に深層学習を適用することは困難である．

２・２　少数サンプルへの対応

少数サンプルにおいても高精度なモデルを作成するため，CG 画像を用いた転移学習を利用する．本研究では，初期パラメータを乱数とした通常の学習，ImageNet を用いた転移学習，CG 画像を用いた転移学習の 3 つの方法を比較して，効果を検証した．

（１）Scratch

通常の学習と評価の流れにおいて少数の学習用サンプルを利用した場合を図 4 に示した．学習/評価用サンプルは２・５節で説明する．(a)パラメータ θ を乱数で初期化し，(b)少数の学習用サンプルでパラメータ θ を θ_1^*に学習し，(c)パラメータ θ_1^*を持つモデルを評価用サンプルで評価した．パラメータ θ から学習する方法を Scratch と呼ぶ. 図 4(b)のように学習用画像が少ない場合は，図 4(c)の評価で良い結果が得られない．

（２）ImageNet Pretrain

ImageNet を用いた転移学習において少数の学習用サンプルを利用した場合を図 5 に示した．ここで 2012 年に開催された大規模画像認識のコンペティションである ImageNet Large Scale Visual Recognition Challenge の画像分類タスクで用いられたデータセット[3)]を ImageNet と表現した．ImageNet は 1,000 クラスの学習用画像 1,281,167 枚，評価用画像 50,000 枚があり，シャム猫等の動物のクラスが多く，金属研削面のキズ有無のクラスはない．(a)乱数で初期化されたパラメータ θ を事前に ImageNet で θ_2に学習し，(b)少数の学習用サンプルでパラメータ θ_2 を θ_2^*に学

習し, (c)パラメータ θ_2^*を持つモデルを評価用サンプルで評価する．ImageNet で事前学習されたパラメータ θ_2 から学習する方法を ImageNet Pretrain と呼ぶ．ImageNet と金属研削面のキズ有/無画像のクラスの分布が大きく異なると考えられるため，学習用画像枚数が少ないときは良い結果が期待できない．

（３）CG Pretrain

CG 画像を用いた転移学習において少数の学習用サンプルを利用した場合を図 6 に示した．CG 画像の生成方法は２・４節で説明する．(a)乱数で初期化されたパラメータ θ を事前に CG 画像で θ_3 に学習し，(b)少数の学習用サンプルでパラメータ θ_3 を θ_3^* に学習し, (c)パラメータ θ_3^*を持つモデルを評価用サンプルで評価する．CG 画像で事前学習されたパラメータ θ_3 から学習する方法を CG Pretrain と呼ぶ．CG 画像と金属研削面のキズ有/無画像のクラスの分布は似ていると考えられるため，学習用画像枚数が少なくても良い結果が期待できる．

２・３　その他の高精度化技術

モデルの精度を決める要因は，データセット，モデル，学習方法であり，転移学習は学習方法の一部である．転移学習以外については PyTorch Image Models（timm）[4] というリポジトリを参考にした．

（１）データセット

現物を撮影する以外に，画像を変形させたり，ラベルを混合したりするデータ拡張により精度を向上させる．本研究では timm train.py の aa オプションに rand-m9- mstd0.5-inc1 を利用した. その他のオプションはデフォルト値とした．

（２）モデル

timm train.py の model オプションに resnet50 を利用した．入力画像サイズは 224×224 画素，3ch，出力は OK と NG の 2 クラスとした．その他のオプションはデフォルト値とした．

（３）学習方法

バッチサイズ 1,024 のとき学習率 0.4 とし，バッチサイズと学習率を比例させた．例えば，バッチサイズ 1 のとき学習率 0.00039 とした．このバッチサイズ 1,024 は全体の数で，例えば Graphics Processing Unit（GPU）を 4 個利用する場合，1GPU あたりのバッチサイズは 256 とした. バッチサイズは 1~1,024 とした．

図７　CG により生成されたキズ有画像の例

図８　現物から撮影されたキズ有画像の例

また，エポック数は 1,000 とした．その他のオプションはデフォルト値とした．バッチサイズによって大きく異なるが，1 回の実験に要する学習と評価の時間は, Tesla V100-SXM2-16GB が 4 枚搭載された GPU サーバで 30 分～6 時間だった．

２・４　CG 画像の生成方法

CG 画像の生成は，金属研削面及びキズをモデリングする形状生成，仮想空間上でその形状に光を当てカメラで撮影する画像化の 2 つに分かれる．

（１）形状生成

研削面及びキズは PyOpenGL を利用して生成した．研削面は円柱をランダムな位置及び向きで，またキズは円柱と球をつなげて描画することで表現した．デプスバッファを利用してカメラから最も近い円柱や球の座標を画素単位で得て，三角形メッシュを生成し，OBJ 形式で保存した．

（２）画像化

次に, OBJ 形式の三角形メッシュを BlenderProc[5]で読み込み，光源及び物体のテクスチャをランダムに変化させる Domain Randomization を利用し，レンダリングし，画像を PNG 形式で保存した．また，物体のマテリアル（色，粗さ，反射率等）のパラメータを金属の質感が失われないようにランダムに決定した．生成された画像を図 7 に示した．

以上の形状生成から画像化までの処理に X virtual framebuffer（Xvfb）を利用し，GPU サーバ上で実行した. CG 画像 1 枚の生成に要する時間は RTX2080Ti が 1 枚搭載された GPU サーバで約 30 秒だった．このため, CG 画像 20,000 枚の生成に約 7 日を要した．

２・５　学習及び評価サンプル

本研究では，解決したい問題を金属研削面のキズ

の有無を判断する 2 クラス分類問題とし，現物から撮影された学習/評価用のキズ有/無画像を各 1,024 枚，計 4,096 枚準備した．撮影された画像を図 8 に示した．キズ有/無の各クラスの学習用画像枚数を 1, 2, 4, 8, ... , 1,024 と倍々に変化させ，少数サンプルにおける精度を調べた．ここで，精度は評価用キズ有/無画像全体（2,048 枚）のうち，正しく分類できた割合で定義される．

3　結果

各クラスの学習用画像枚数を 1 から 1,024 まで変化させたときの Scratch，ImageNet Pretrain，CG Pretrain の精度を表 1 に示した．例えば，各クラスの学習用画像枚数 1 は学習用のキズ有画像が 1 枚，キズ無画像が 1 枚，計 2 枚でモデルを学習し，2,048 枚の評価用キズ有/無画像で精度評価したということを意味する．このとき Scratch の精度は 55.71%，つまり 2,048 枚中 1,141 枚が正解だった．ImageNet Pretrain は，Scratch と比較して，各クラスの学習用画像枚数 1~1,024 の全てで精度向上が確認できた．CG Pretrain は，Scratch 及び ImageNet Pretrain と比較して，各クラスの学習用画像枚数 128 までは最高精度で，それ以降は同等程度の精度だった．この結果から CG Pretrain は各クラスの学習用画像枚数が 128 枚まで，特に少ない枚数ほど有効であるといえる．

また，DoG フィルタ処理，2 値化，ラベリング，各ラベルのキズ判定といった画像処理を適用したときの精度は 95.20%だった．深層学習を利用すると画像処理では達成できなった精度が少数サンプルにおいても容易に達成できることがわかった．

4　まとめ

金属研削面のキズの有無を判断する 2 クラス分類問題において，各クラスの学習用画像枚数を 1 から 128 まで変化させたとき，モデルの初期パラメータを乱数とした通常の学習及び ImageNet を用いた転移学習のモデルと比較して，CG 画像を用いた転移学習のモデルがより高精度だった．また，深層学習を利用すると画像処理では達成できなった精度が少数サンプルにおいても容易に達成できることがわかった．

表 1　精度の比較

各クラスの学習用画像枚数	精度[%]		
	Scratch	ImageNet Pretrain	CG Pretrain
1	55.71	68.46	99.23
2	54.30	89.50	99.23
4	57.62	92.14	99.23
8	57.57	98.24	99.23
16	59.67	99.17	99.41
32	60.01	99.32	99.81
64	85.06	99.51	99.76
128	98.68	99.71	99.85
256	99.76	99.85	99.76
512	99.76	99.85	99.85
1,024	99.90	99.95	99.90

参考文献

１）Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell. “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition”. Proceedings of the 31st International Conference on Machine Learning, PMLR. 2014, Vol.32, No.1, p.647-655.

２）Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy,　Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, Stan Birchfield. “Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization”. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2018, p.1082-1090.

３）Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge”. International Journal of Computer Vision. 2015, Vol.115,No.3,p.211-252.

４）Ross Wightman. “PyTorch Image Models”. https://github.com/rwightman/pytorch-image-models. (accessed 2022-08-26)

５）Maximilian Denninger. “BlenderProc”. https://github.com/DLR-RM/BlenderProc. (accessed 2022-08-26)