

機械学習を用いたコンクリートのひび割れの自動検出器の開発

Development of an automatic detector of cracks in concrete using machine learning

北海道大学工学部環境社会工学科 ○学生員 横山 傑 (Suguru Yokoyama)
北海道大学大学院工学研究院 正 員 松本高志 (Takashi Matsumoto)

1. はじめに

近年、日本の建造物の老朽化が目立ってきており、効率的な維持管理が重要となっている。こうした中、コンクリートのひび割れを建造物の写真から自動検出する検出器は効率的な維持管理に貢献すると考えられる。

物体検知は通常、画像解析と機械学習を組み合わせで行われる。あらかじめ目標となる対象の特徴パターンをコンピューターに学習させておき、対象が含まれる画像を入力されたとき、学習されている特徴パターンと照らし合わせて判定を行う。

現在、画像認識の分野においては深層学習が注目されている。深層学習とは、脳の神経回路網を模倣した人工ニューラルネットワークの層を多層にしたものを用いた学習方法である。深層学習を用いた画像認識技術は、画像認識のベンチマークテストにおいて、過去の記録を次々に塗り替えるようになり、その有用性が認められるようになった。

以上を踏まえ、画像認識に深層学習を用いることにより、コンクリート上におけるひび割れの十分な検出精度を期待できると考えた。深層学習の手法は数多く提案されているが、その中でも最も画像認識に特化した、畳み込みニューラルネットワークを用いるものを利用する。本研究では、自動検出器の開発に必要な、入力された画像がひび割れであるかどうかを分類する、畳み込みニューラルネットワークを構築し、分類精度を検討する。

2. ネットワークの構成

本章では、構築する畳み込みニューラルネットワークの各層の働きを説明する。ここでは参考文献 1)を参考にしている。今回学習させた畳み込みニューラルネットワークの畳み込み層、プーリング層、全結合層、識別層は図-1 のように接続される。学習は誤差逆伝搬法と確率的勾配降下法により学習データと正解との、式(1)の誤差関数を最小化するように重みを更新していくことで行われる。誤差関数とは学習データに対する重みの尤度に対数を取り符号を反転したものである。なお、入力画像は $64 \times 64 \text{pixel}$ の 3 チャンネルカラー画像であるため、画像のサイズを $64 \times 64 \times 3$ と表すこととする。また、中間層での出力は画像と呼ばず、マップと呼ぶ。

第 l 層での計算は、式(2),(3),(4)のように表される。

$$E(w) = - \sum_{n=1}^N \sum_{k=1}^K d_{nk} \log y_k(x_n; w) \quad (1)$$

$$u^{(l)} = w^{(l)} z^{(l-1)} + b^{(l)} \quad (2)$$

$$f(u) = \max(0, u) \quad (3)$$

$$z^{(l)} = f^{(l)}(u^{(l)}) \quad (4)$$

ここに、 d :正解、 x :学習データ、 N :学習データ数、 K :分類クラス数、 u :入力値、 w :重み、 z :出力値、 b :バイアスである。

2.1 畳み込み層

フィルタと呼ぶサイズの小さい画像を考え、フィルタ

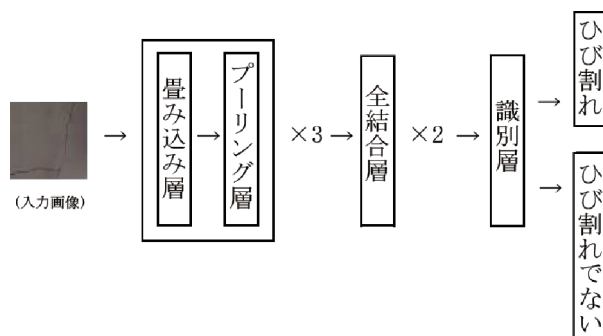


図-1 畳み込みニューラルネットワーク

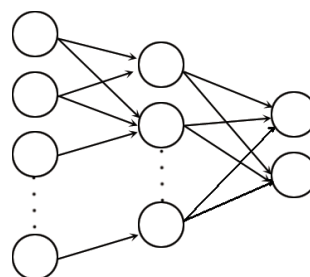


図-2 畳み込み層

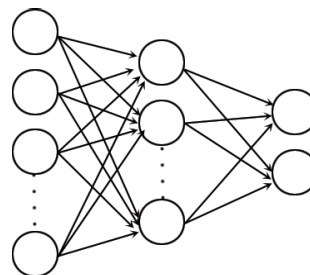


図-3 全結合層

の濃淡パターンと類似した濃淡パターンが入力画像上のどこにあるのかを検出する、つまりフィルタが表す特徴的な濃淡構造を、画像から抽出する働きがある。脳の視覚野をヒントにしている、図-2 のように隣接層間の特定ユニットとのみ結合をする。これにより、画像から抽出される特徴の次元が小さくなるため、効率的な学習が可能となっている。

2.2 プーリング層

畳み込み層で抽出された特徴の位置感度を若干低下させることで、特徴の微小な位置変化に対して不変な出力を行う。畳み込み層同様に隣接層間の特定ユニットのみと結合をする。この層の結合の重みは調節可能なものではなく、固定されているため、プーリング層には学習によって変化するパラメータはない。

2.3 全結合層

畳み込み層で抽出された特徴量を学習する。畳み込み層やプーリング層と違い、図-3 のように隣接層のユニットすべてが全結合をする。

2.4 識別層

全結合層から出力された情報をソフトマックス関数に通すことで、入力画像の所属がどのクラスであるか、それぞれのクラスに対し確率を出力する。このとき確率が最大となるクラスに、入力された画像は分類される。ソフトマックス関数は式(4)で表される。

$$z_k^{(L)} = \frac{\exp(u_k^{(L)})}{\sum_{j=1}^K \exp(u_j^{(L)})} \quad (4)$$

ここに、 k :出力層のユニット番号である

3. 学習データセット

まず、図-4 のような、ひび割れ部分を含むコンクリートの写真を実際に撮影し、またインターネット上からフリー素材を集めたりすることで 287 枚収集した。図-4 は北海道大学工学部内の壁の写真である。このような写真からひび割れのみ写っている部分を 128*128pixel で 1479 枚、コンクリートの打ち継ぎ目や、何もないところや、影、チョークの線などの、ひび割れが写っていない部分を同様にして 1472 枚、図-5、6 のように切り出した。これらを、鏡像反転、回転、四隅を 100*100pixel で切り取るなどして、それぞれ図-7 のように 11832 枚、14720 枚にデータ拡張した。最後に、64*64pixel にリサイズして、これを学習のデータセットとする。なお、学習されたネットワークを評価するためのテストデータを、それぞれ 224 枚、355 枚集めた。

4. 学習と評価方法

図-8 のように、学習データの画像から画像のすべての画素の RGB 平均値を引くという前処理をした場合と、図-7 のように何も前処理をしない場合を考える。



図-4 ひび割れ写真

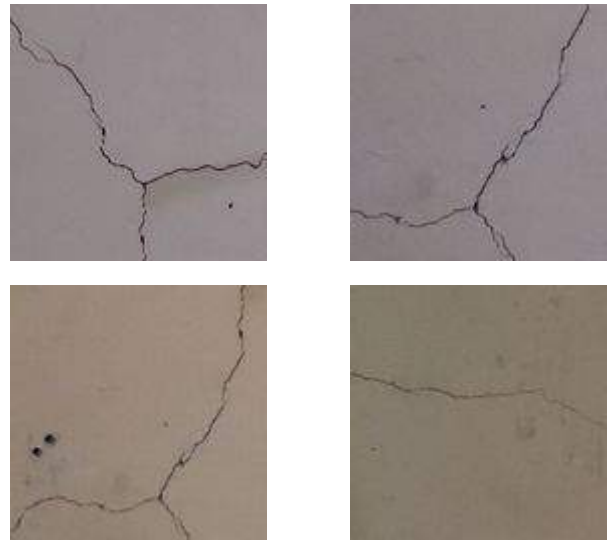


図-5 ひび割れ部分(128*128pixel)



図-6 ひび割れでない部分(128*128pixel)

学習を効率化するため、学習データを 50 枚読み込むごとに重みの更新を行うミニバッチ学習を用いる。ネットワークが学習データに過適合して汎化性能を失うのを防ぐために、ネットワークの自由度を強制的に小さくする。具体的には、重みの大きさの上限を 0.5 に制約し、また、ネットワークのユニットを 50%の確率で無効化

しドロップアウトをする。畳み込み層と全結合層の活性化関数には、式(3)の $\text{ReLU}^2)$ を用いる。 $\max(0, u)$ という性質上、多くの値がちょうど0になり、出力がスパースになる。そのため ReLU を用いたネットワークはいくつかのノードのみが非零であるようなスパースなネットワークになる。また、学習係数の調整には Adam²⁾ を用いる。Adam は勾配の平均と分散をオンライン推定し、これらの情報を利用して学習率を更新する手法である。

学習は 20 回繰り返しを行い、学習終了ごとに学習データ、テストデータを入力し、分類精度を記録する。なお、学習データの分類精度を訓練精度、テストデータを用いた分類精度をテスト精度と呼ぶこととする。各層のパラメータは表-1 に示す。

5. 開発環境

本研究の開発環境は表-2 に示す。一般的な PC で開発している。

6. 評価結果

学習データに前処理をしていない場合、図-9 のように、訓練精度は学習を進めるごとに順調に向上しているのに対し、テスト精度は epoch6 の 70.5%をピークとしてこれ以上向上することはなかった。なお、学習には 1401 秒かかった。

学習データに前処理をした場合、図-10 のように、テスト精度は 78.9%に向上したが、これも epoch4 をピークにこれ以上上昇することはなかった。なお、学習には 1389 秒かかった。

今回最も高い分類精度を示した、学習データに前処理をした場合の epoch4 での分類の結果、正確に分類された画像の例を図-11、12 に、誤って分類された画像の例を図-13、14 に示す。図-11 のようにひび割れのみが写った画像や、図-12 のように何も写っていない画像やチョークの線が写っている画像などは正確に分類されることが多かった。一方、図-13 のように、ひび割れの上に影が映っている場合や、直線状にひび割れている場合は、ひび割れでない方に分類されてしまうことが多かった。また、図-14 のような、コンクリートの打ち継ぎ目は、ひび割れに誤分類されてしまうことが多かった。

7. 考察と今後の課題

今回、図-10 のように、学習を進めるごとに学習データの分類精度は上昇しているのに対し、テスト精度は epoch4 をピークに向上していない。このことから epoch5 以降、ネットワークが学習してほしい特徴以外の特徴も学習してしまい、学習データに対し過適合してしまっているため、汎化性能を失い、訓練精度とテスト精度に大きな差ができてしまっていることがわかる。ネットワークが学習データに過適合してしまう原因としては学習データの数が少ないことが考えられる。また、学習データに前処理を行わない場合、図-9、10 のように、



図-7 拡張されたデータ(100*100pixel)



図-8 図-7 に前処理をした画像(100*100pixel)

表-1 ネットワークパラメータ

層種	パッチ	ストライド	出力マップサイズ	関数
data	-	-	64×64×3	-
conv1	5×5	1	64×64×32	ReLU
pool1	3×3	2	32×32×32	-
conv2	5×5	1	32×32×64	ReLU
pool2	3×3	2	16×16×64	-
conv3	5×5	1	16×16×64	ReLU
pool3	3×3	2	8×8×64	-
fc4	-	-	1×1×512	ReLU
fc5	-	-	1×1×2	softmax

data:入力画像 conv:畳み込み層 pool:プーリング層
fc:全結合層

表-2 開発環境

OS	windows7 (64ビット)
CPU	Corei5-4460 3.2GHz
RAM	8GB
GPU	GTX750Ti
言語	python3
IDE	PyCharm

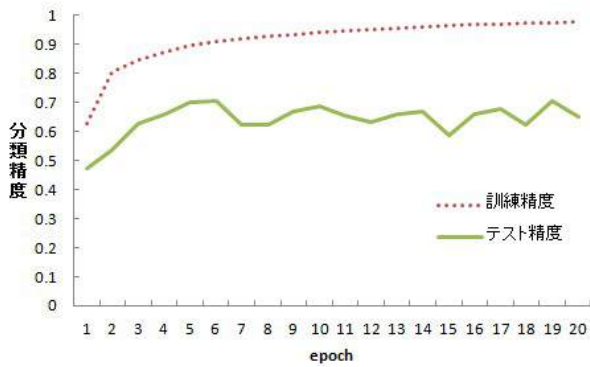


図-9 前処理なしの評価結果

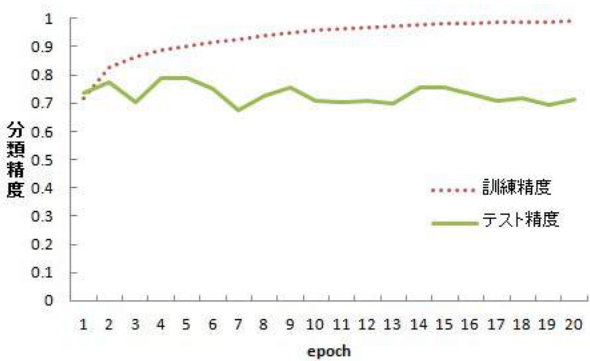


図-10 前処理ありの評価結果

前処理をした時と比べてテスト精度が下がるため、なんらかの方法で学習データに前処理をすることで精度の改善を見込めると考えられる。そのため、学習データの数を増やす、様々な前処理を検討するなどが今後の課題である。今回、一般的 PC で開発していたためスペックの関係上、学習データ数や出力マップサイズなどのパラメータを大きな値に設定することができなかった。そのため、今後は Amazon の仮想クラウドサーバーである Amazon EC2 の GPU インスタンスを利用することでスペックの不足分を補おうと考えている。

十分な分類精度を確保できたら、図-4 のようなひび割れが含まれる一枚のコンクリートの写真を入力したとき、写真上に矩形をラスタスキャンして画像を切り取り、学習させた畳み込みニューラルネットに入力させることで、その画像がひび割れであるかそうでないかを分類し、ひび割れと判定された部分を写真上に表示させるというコンクリートのひび割れ自動検出器を開発する。

8. おわりに

本論文のまとめを以下に示す。

構造物の効率的な維持管理を目指して、コンクリートのひび割れの自動検出器を開発している。自動検出器の開発には、まず画像がひび割れであるかないかを正確に分類する分類器が必要である。分類器には、高い分類精度を期待できる畳み込みニューラルネットを使用した。ネットワークに学習させるデータになんらかの前処理を施すことで分類精度が向上することがわかった。今後、



図-11 正確に分類されたひび割れ画像(128*128pixel)

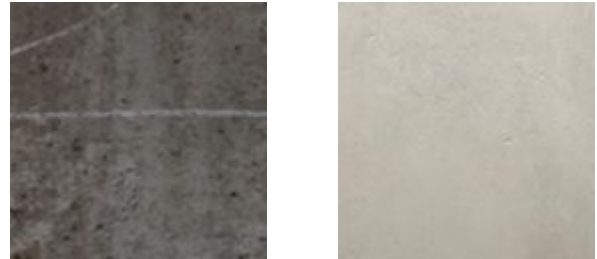


図-12 正確に分類されたひび割れでない画像(128*128pixel)



図-13 誤分類されたひび割れ画像(128*128pixel)

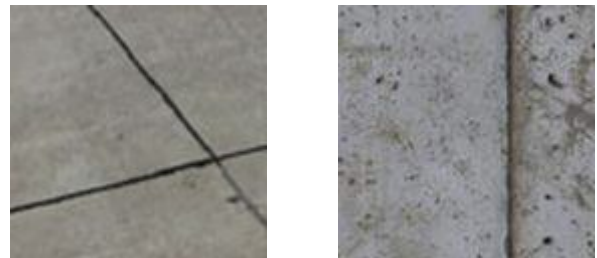


図-14 誤分類されたひび割れでない画像(128*128pixel)

学習データを増やす、学習データにいろいろな前処理を施すことで更なる分類精度の向上を図る。満足する分類精度が得られたのち、ひび割れがあるコンクリートの写真からひび割れ部分を検出する検出器を開発することとする。

参考文献

- 1) 岡谷貴之: 機械学習プロフェッショナルシリーズ、深層学習、講談社、2015
- 2) 人工知能学会: 深層学習、近代科学社、pp.143-145、2015