

Deep Learning によるコンクリートのひび割れ自動検出器の開発と SNS を用いたシステムの実装

Development of an automatic detector of cracks in concrete using Deep Learning and the system implementation using SNS

北海道大学工学院
北海道大学大学院工学研究院

○学生員 横山 傑 (Suguru Yokoyama)
正 員 松本高志 (Takashi Matsumoto)

1. 序論

近年、日本の構造物の老朽化が進むとともに増えてきており、効率的な維持管理が重要となってきた。特に、コンクリート構造物におけるひび割れは、劣化損傷要因を間接的に表すとともに、耐久性の観点から弱点となるものであるため、その長さ、幅、範囲などを点検において取得することは大変重要である。しかしながらその量は膨大であるため、コンクリートのひび割れを構造物の写真から自動検出する検出器が開発されたならば、当分野の効率的な維持管理に大きく貢献するものと考えられる。

物体検出は通常、画像処理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が入力されたとき、学習されている特徴パターンと照らし合わせることで判定を行う。

現在、各分野において、Deep Learning の認識精度の高さが注目されている。Deep Learning 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に自動特徴抽出層を追加したものをを用いた学習方法である。コンピュータービジョンの分野では特に Deep Learning である深層畳み込みニューラルネットワーク (CNN; Convolutional Neural Network)¹⁾が高精度な物体認識に貢献している。ImageNet という 100 万枚を超える画像のデータセットから 1000 クラスの分類を行って分類精度を競う ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 において、CNN を用いた手法が過去の記録の認識率 25.8% を 16.4% に大きく塗り替え、その有用性が認められるようになった²⁾。

以上を踏まえ、画像認識に CNN を用いることにより、コンクリート表面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。本研究では、コンクリート表面の写真からひび割れ、エフロレッセンス(以後、エフロ)、チョーク文字を検出可能な CNN による自動検出器を開発した。さらに、Twitter を用いてひび割れ自動検出器アカウント (@cracks_detector) に変状などを含むコンクリート表面の写真を添付しリプライすることで、検出結果が返ってくるシステムを構築した。

2. 畳み込みニューラルネットワークの構成

本章では、参考文献³⁾⁴⁾を参考にして構築する CNN の各層の働きを説明する。今回学習させた CNN の特徴抽

出層(畳み込み層、プーリング層)、全結合層、識別層は図-1(右)のように接続される。学習は誤差逆伝搬法⁵⁾と Adam⁶⁾により学習データの出力と正解ラベルとの誤差関数である、式(1)の交差エントロピー誤差を最小にするようにノード間の重みやバイアスを更新していくことで行われる。なお、入力画像は 128[pixels]×128[pixels]のグレースケール 1[channel]であるため、画像のサイズを 128×128×1 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。

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

$$u_i^{(l)} = \sum_{j=1}^J w_{ij}^{(l)} z_j^{(l-1)} + b_i^{(l)} \quad (2)$$

$$f(u) = \max(0.2u, u) \quad (3)$$

$$z_i^{(l)} = f(u_i^{(l)}) \quad (4)$$

ここで、 E :交差エントロピー誤差、 n :学習データ番号、 N :学習データ数、 k :クラスラベル、 K :分類クラス数、 d :教師ラベル、 y :ネットワークの出力値、 u :活性化関数の入力値、 l :層番号、 i : l 層のノード番号、 j : $l-1$ 層のノード番号、 J : $l-1$ 層のノード数、 w :重み、 z :活性化関数の出力値、 b :バイアスである。

2.1 畳み込み層

図-1(左)のような重みフィルタと呼ぶサイズの小さい画像を考え、フィルタの濃淡パターンと類似した濃淡パターンが特徴マップ上のどこにあるのかを畳み込み演算により検出する。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。これは脳の視覚野をヒントにしておき、図-2(左)のように隣接層間の特定ユニットとのみ結合をする。特徴マップ上にフィルタを走査する間隔をストライドと言い、本研究ではストライドを 1 としている。学習時はフィルタの重みとバイアスを更新する。

2.2 Max Pooling 層

畳み込み層で抽出された特徴マップを参照し、図-3のように局所的な最大値を取り出して位置感度を若干低

2	0	1
0	1	2
1	0	2

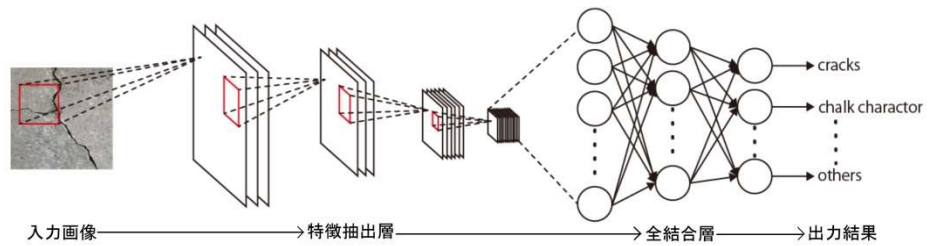


図-1 重みフィルタ(左)と畳み込みニューラルネット(右)

下させることで、特徴の微小な位置変化に対してロバストな出力を行う。また、特徴マップから抽出される特徴の大きさが小さくなるため、効率的な学習が可能となる。このように Max Pooling 層は、局所領域から最大値を取るだけの処理なので、学習によって変化するパラメータはない。

2.3 Spatial Pyramid Pooling 層

SPP(Spatial Pyramid Pooling)⁷⁾層は特徴マップを格子状に 1、4、16、25 と分割していき、その中でそれぞれ Max Pooling を行う。その後、各領域を繋げたベクトルを出力する。したがって Max Pooling 層と違い、入力サイズによらず固定サイズの出力を行う。本研究では最後の Pooling 層に SPP を用いている。

2.4 全結合層

特徴抽出層(畳み込み層とプーリング層)で抽出された三次元の特徴データを一次元のベクトルデータに変換し学習をする。畳み込み層やプーリング層と違い、図-2(右)のように隣接層のユニットすべてが全結合をする。また、第1層での計算は、式(2),(3),(4)のように表される。学習時はノード間の重みとバイアスを更新する。

2.5 識別層

全結合層から出力されたデータ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(5)で表される。

$$z_k = \frac{\exp(u_k)}{\sum_{k=1}^K \exp(u_k)} \quad (5)$$

3. 学習データセット

まず、図-4 のような、変状部分を含むコンクリート表面の写真を 2000 枚用意した。このような写真から硬いひび割れ、柔らかいひび割れ、ひび割れ付きエフロ、エフロ、チョーク文字、何もない表面、打ち継ぎ目、さび、その他部分すべての 9 クラス(図-5)を 128[pixel]×128[pixel]のサイズで図-5 の枚数を収集した。これらを、鏡像反転、上下反転、90° 回転により 4 倍にデータ数の拡張を行い、これを学習のデータセットとする。また、学習されたネットワークを評価するため、学習データセットとは別に、テストデータとして、ひび割れ、チョーク、表面、打ち継ぎ目、その他すべての 4 クラスを各

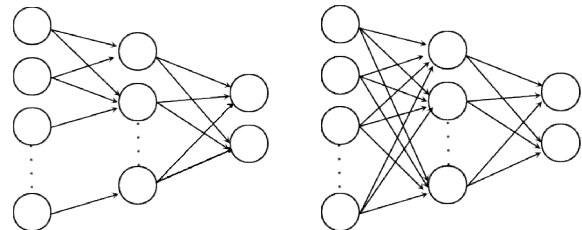


図-2 畳み込み層(左)と全結合層(右)

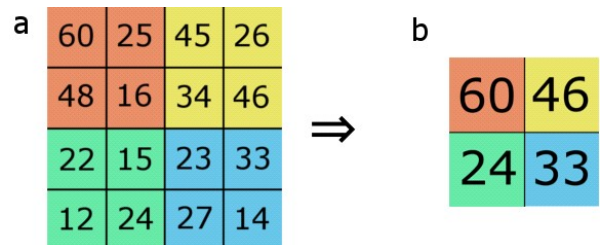


図-3 Max Pooling



図-4 コンクリート表面の写真

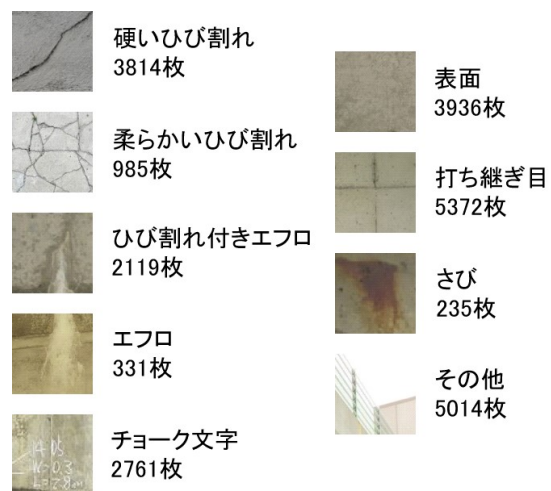


図-5 学習データ

1000 枚ずつ用意し評価用データセットとした。なお、本タスクにおいてカラー情報は不要であると考え、デー

タセットはすべてグレースケール化している。

4. 学習と評価方法

学習データの画像において、グレースケール画像のすべての画素の平均値を引き、画素の平均が 0 となる前処理をする。

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、ネットワークのユニットを確率的に無効化し dropout⁸⁾をする。畳み込み層と全結合層の活性化関数には、式(3)の Leaky ReLU⁹⁾を用いる。また、学習係数の調整には Adam を用いる。Adam は損失関数の値をできるだけ小さくするようにネットワークのパラメータを最適化する手法である。

学習は 200 回繰り返して行い、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. 分類器の学習と評価結果

分類器は AWS(Amazon Web Services)の g2.xlarge にて GPU を用いて学習させる。OS は ubuntu、言語は python3、機械学習フレームワークには chainer を用いる。

学習された分類器の評価結果を図に示す。訓練精度が伸びなくなり始めた、学習 150 回目の分類器を用いることとする。なお、この分類器のテスト精度は 82%である。学習時間は 45 時間 6 分 47 秒であった。

6. ひび割れ自動検出器の実装

学習した分類器を用いて自動検出器を実装する。入力画像は、撮影対象から 1~3m の位置から撮影された解像度 1K~4K の物を想定している。入力された画像は、長辺が 1920pixel 以上なら 1920pixel に、1080pixel 以下なら 1080pixel に縦横比を維持しつつリサイズされる。入力画像上にて、130[pixel]×130[pixel]、110[pixel]×110[pixel]、90[pixel]×90[pixel]の 3 種類のスライドウィンドウをそれぞれ 80[pixel]、65[pixel]、50[pixel]の間隔で図-7 のようにラスタスキャンさせる。随時スライドウィンドウ上の画像を 128[pixel]×128[pixel]にリサイズしグレースケール化した後、学習済みの分類器で検出対象とそうでないものに分類する。ひび割れ、エフロ、チョーク文字に分類された場所を入力画像にそれぞれ赤、青、緑で塗りつぶして表示することで、検出結果を示す。

7. 自動検出器を用いた検出結果と考察

ひび割れとエフロのあるコンクリートの写真(600[pixel]×800[pixel])からひび割れとエフロを検出した結果を図-8 に、エフロとチョーク文字のあるコンクリートの写真(600[pixel]×800[pixel])からエフロとチョーク文字を検出した結果を図-9 に示す。なお、図中の白枠内の画像が全体像である。検出結果はそれぞれ 15 秒であった。図の左が正解であり、右が検出結果である。

図には打ち継ぎ目が含まれているが、誤検出はされていない。図-9 の下部にひび割れが誤検出されている。これは学習データのひび割れ画像にチョークの矢印が混在することがあったため、チョークの矢印をひび割れと誤検出する可能性がある。しかし目印として描かれているこの矢印の横にはひび割れがあることが多いので考慮する問題ではないと考えられる。また、打ち継ぎ目に見られる通り、写真は真正面から撮られたものではなく斜めに傾いているが問題なく検出されており、対象物の傾きに対するロバスト性を持つことがわかる。他の検出結果の例を図-10 に示す。

表-1 ネットワークパラメータ

層種	バッチ	ストライド	出力マップサイズ	関数	dropout
data	-	-	128×128×1	-	-
conv1	3×3	1	128×128×32	LReLU	-
conv2	3×3	1	128×128×32	LReLU	-
pool2	2×2	2	64×64×32	-	0.2
conv3	3×3	1	64×64×64	LReLU	-
conv4	3×3	1	64×64×64	LReLU	-
pool4	2×2	2	32×32×64	-	0.2
conv5	3×3	1	32×32×96	LReLU	-
conv6	3×3	1	32×32×96	LReLU	-
pool6	-	-	1×1×8160	-	0.2
fc7	-	-	1×1×1024	LReLU	0.5
fc8	-	-	1×1×9	softmax	-

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

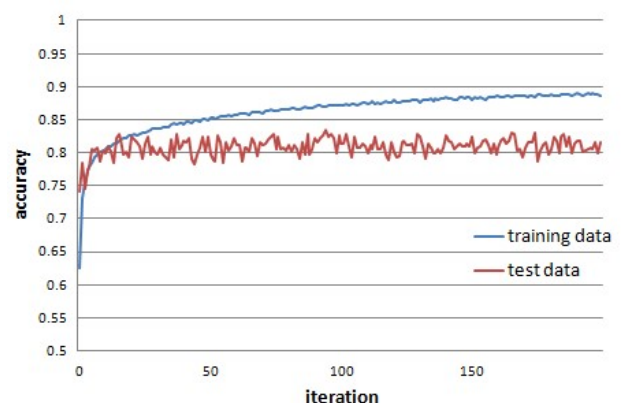


図-6 学習結果

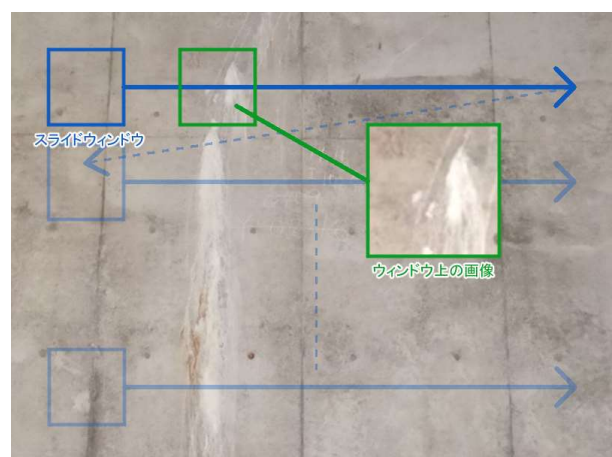


図-7 スライドウィンドウ

8. Twitter を用いた自動検出システムの構築

自動検出器は python で書かれているため、このままでは使用者のパソコンに専用の環境を構築する必要がある。そこで、Twitter¹⁰⁾にコンクリートの写真を添付して、ひび割れ自動検出器アカウント(@cracks_detector)にリプライを飛ばすと、検出結果が返ってくるというシステムを構築した。システムのフローは図-11 の通りである。Twitter に添付された写真は、北海道大学工学部構造デザイン工学研究室にあるサーバーに送られ、サーバーにより計算され結果が出力される。出力された検出結果はリプライに添付されて返ってくる仕組みとなっている。

9. 結論

本研究では構造物の効率的な維持管理が重要であると考え、コンクリートの写真からひび割れ、エフロ、チョーク文字を自動検出できる検出器を開発した。また、誰でも簡単に本検出器を利用できるようにするため、サーバーとの通信に Twitter を用いたシステムを構築した。自動検出器の作成には、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、82%の分類精度が確認された。開発された自動検出器は誰でも簡単に使用することができる。

将来展望として、検出精度の向上、検出対象の種類拡張が必要であると考えている。

参考文献

- 1) Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel : Backpropagation applied to handwritten zip code recognition, Neural Computation, Vol.1, pp.541-551, 1989.
- 2) A. Krizhevsky, I. Sutskever and G. Hinton : ImageNet classification with deep convolutional neural networks, NIPS, 2012.
- 3) 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.
- 4) 人工知能学会: 深層学習、近代科学社、pp.143-145、2015.
- 5) D. E. Rumelhart, G. E. Hilton, and R. J. Williams : Learning representations by back-propagating errors, Nature Vol.323, No. 6088, pp533-536, 1986.
- 6) D. Kingma and J. Ba : Adam: a method for stochastic optimization, arXiv:1412.6980, 2014.
- 7) K. He, X. Zhang, S. Ren, and J. Sun : Spatial pyramid pooling in deep convolutional networks for visual recognition, arXiv:1406.4729v4 [cs.CV], 2015.
- 8) N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov : Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research, pp1929-1958, 2014.
- 9) A. L. Maas, A. Y. Hannun, and A. Y. Ng : Rectifier nonlinearities improve neural network acoustic models, 30th International Conference on machine Learning, 2013.
- 10) Twitter, Inc : Twitter: <http://twitter.com>.

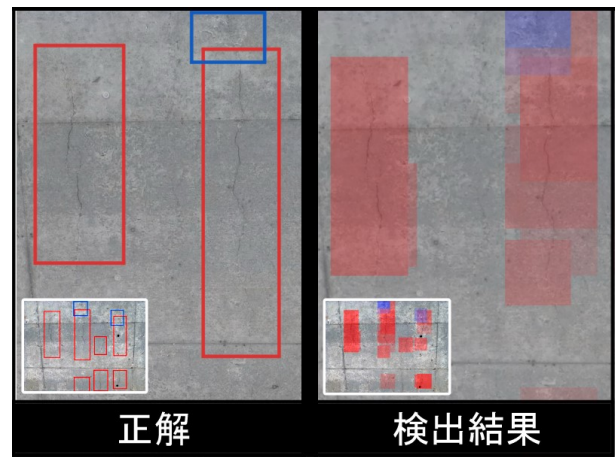


図-8 ひび割れとエフロ検出結果

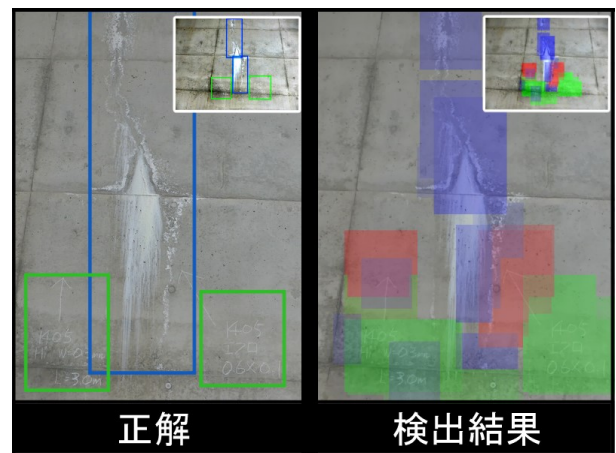


図-9 エフロとチョーク文字検出結果



図-10 検出結果

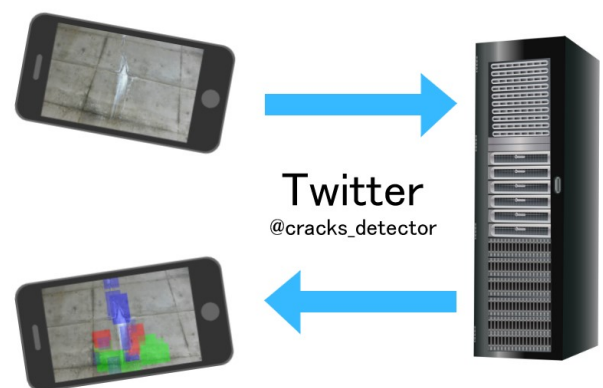


図-11 自動検出器のシステムフロー