

深層学習を用いたインプラントの画像識別

平川智喜¹ 松下恭之² 荒平高章¹

概要: 歯科医療の問題である埋入されたインプラントの種類の特定に対して、深層学習を用いて識別を目指した。具体的な手法は、深層学習の畳み込みニューラルネットワークを用いており、学習用データには4種類のインプラントのレントゲン画像を用いた。また学習用データは4種類と2種類(6パターン)のインプラントに分けてそれぞれ学習させた。その結果、4種類の正答率は61.7%、2種類の正答率は57.9%~100%を示した。本研究では課題が多く見つかった。今後の展望として、開発環境の向上や多くのインプラントを学習させることで歯科医療での実用化を目指す。

キーワード: AI, Implant, CNN, Deep learning, Image identification.

1. はじめに

歯科における臨床問題として、施術やメンテナンスの際に、患者に埋入されたインプラントの種類が判別できないことがある。現在インプラントの種類判別は、レントゲン画像から臨床医が文献や経験に基づいて行っている。インプラントはメーカーによって素材や形状、構造といった規格が違い、施術する上で必要な器具が異なっている。インプラントはメーカーだけでも世界中で100社以上、日本国内で30社以上も使われており、ベテランの臨床医でも的確な鑑別は難しい。この問題の解決策として、人工知能の一種である深層学習を用いる。深層学習は、医療や製造業、農業、自動車といった分野における画像認識や音声認識を得意としている。本研究では、深層学習の一種である畳み込みニューラルネットワークを構築し、4種類のインプラントの画像を識別することで、どのインプラント同士が識別できるか明確化することを目的とする。

2. 理論

深層学習は、神経細胞を模倣したニューラルネットワークから構築されている。ニューラルネットワークは入力層、中間層、出力層の3つから主に構成されており、中間層が2層以上のモノを深層学習と呼ぶ。この層を深くすることで、画像認識や音声認識といった複雑な問題を解くことができる。ニューラルネットワークはニューロンの集まりから構成されている。入力層の値に重みを掛けて中間層へ伝播させる。中間層を深くするときに必要な関数が活性化関数である。活性化関数は非線形で、閾値を境に入力層から伝播した値を変化させる関数である。こうして出力層に送られてきた値は、ソフトマックス関数により確率として予測値を出力する。確率として出力された値が正しいかを測るため、損失関数を用いる。損失関数は予測値と教師データが合っているかを確認する。もし予測値と教師データが

合っていない場合、最適化関数で重みを更新する。この重みを更新しながら損失関数の値を最小にしていく。またニューラルネットワークを学習していくと、学習データに過剰に学習してしまうことがある。人工知能は未知のデータに対して識別できることが目的であるため、学習データだけを識別できても意味がない。その対策としてドロップアウト層を設ける[1]。ニューロンの一部をなくすことで、学習データに過剰に学習してしまう現象である過学習の抑制になる。

3. 材料と方法

3.1 データ

本研究の学習データには、文献[2]に記載されているインプラントのレントゲン画像を用いた。具体的には、米国BioHorizons社のインターナルインプラント、エクスターナルインプラント、レーザーロックテーパード、ノーベルバイオケア社のノーベルアクティブ3.0の4種類である。各インプラントのレントゲン画像を図3.1に示す。

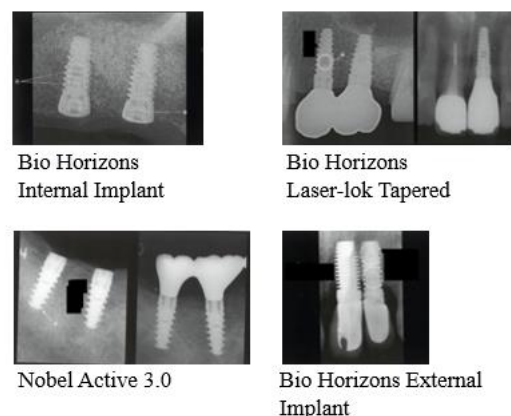


Figure 3.1 Training data.

¹ 九州情報大学
Kyushu Institute of Information Sciences
² 九州大学病院 インプラントセンター
Kyushu University Hospital Implant Center

4枚の画像から計22個のインプラントを切り出した。各データの詳細については表3.1に示す。

Table 3.1 Number of data.

Id	Name	Number	After pretreatment
0	Nobel Active 3.0	8	9016
1	Bio Horizons Internal Implant	4	4508
2	Bio Horizons Laser-lok Tapered	6	6762
3	Bio Horizons External Implant	4	4508

本研究では、4種類のインプラントと2種類のインプラントに分けて、それぞれ学習させる。これにより、4種類の学習がうまくいかなかった際に、どのインプラント同士が識別できているかを正確に確認できる。

3.2 前処理

前節で切り出したインプラントのレントゲン画像に対して前処理を行う。前処理は、人工知能にデータを学習させる前に、データを人工知能が学習できる形に整形することである。データ分析では前処理が8割と言われるぐらい大事な処理になる。本研究では前処理として、回転(-80~80度)、拡大・縮小(0.7~1.3倍)、画像サイズの統一(100×100×3)、正規化(0~255を0~1の範囲へ)を施した。前処理の詳細な過程を図3.2に示す。

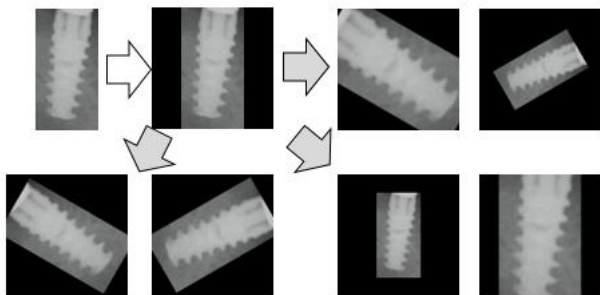


Figure 3.2 Preprocessing.

全データの80%を学習用データ、学習用データの20%を検証用データ、全データの20%をテスト用データとした。また学習用データはネットワークの重みを更新するデータ、検証用データはネットワークの精度を検証するデータ、テスト用データはネットワークの学習後にネットワークの汎用性をテストするデータとして使用する。検証用データとテスト用データは学習に使用しない。前処理を施した学習データの数は、4種類の識別に使用する学習用データは19159枚、テスト用データは5635枚、2種類の識別に使用する学習データの詳細は表3.2に示す。

Table 3.2 Two types and Four types of learning data.

Id.1	Id.2	Train	Test
0, 1, 2, 3		19159	5635
0	1	10143	3381
0	2	12397	3381
0	3	10143	3381
1	2	9016	2254
1	3	6762	2254
2	3	9016	2254

3.3 学習モデル

本研究の開発環境は、Google社が提供しているGoogle Colaboratoryを利用した。学習モデルは、文献[3]を参考に作成し、畳み込みニューラルネットワークを採用している。構成については図3.3に示す。

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 98, 98, 32)	896
max_pooling2d (MaxPooling2D)	(None, 49, 49, 32)	0
conv2d_1 (Conv2D)	(None, 47, 47, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 23, 23, 64)	0
conv2d_2 (Conv2D)	(None, 21, 21, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 10, 10, 128)	0
flatten (Flatten)	(None, 12800)	0
dropout (Dropout)	(None, 12800)	0
dense (Dense)	(None, 512)	6554112
dense_1 (Dense)	(None, 2)	1026

Total params: 6,648,386		
Trainable params: 6,648,386		
Non-trainable params: 0		

Figure 3.3 Configuration of the neural network used in the study.

畳み込みニューラルネットワークは、畳み込み層とプーリング層から構成されており、画像の形状を特徴として学習することができる。畳み込み層では、入力画像と格子状のフィルタを行列積で計算し、特徴マップを生成する。プーリング層では、特定の領域内での最大値や平均値を出力することで、入力データの微小なズレにも同じ値を返すようになり、特徴マップが頑健になる。

3.4 学習モデルの評価

この学習モデルを評価するために、MNISTの0~9の手書き数字のデータセットを用いて学習させた。データの詳細については、学習用データが54,000枚、検証用データが6,000枚、テスト用データが10,000枚である。またデータサイズは28×28×1である。学習した結果、学習中の正答率は99.5%、損失は0.01。検証用データに対する正答率は

は 99.1%, 損失は 0.03. テスト用データに対する正答率は 99.1, 損失は 0.02 と非常に高い精度を示したため, 使用するニューラルネットワークに問題がないことが分かった.

4. 結果と考察

4 種類のインプラントの学習結果を表 4.1, 2 種類の学習結果を表 4.2 に示す.

Table 4.1 Learning results of four types of implants.

Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
0.006	0.998	1.241	0.451	0.999	0.617

Table 4.2 Learning results of two types of implants.

Id.1	Id.2	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
0	1	0.002	0.999	0.000	1.000	0.000	1.000
0	2	0.003	0.999	0.008	1.000	0.016	0.991
0	3	0.001	1.000	0.008	1.000	0.069	0.975
1	2	0.051	0.984	7.404	0.029	3.029	0.579
1	3	0.001	1.000	0.153	0.963	0.022	0.992
2	3	0.000	1.000	0.023	0.984	0.236	0.922

4 種類のインプラントを学習させた結果, テスト用データに対する正答率は 61.7%, 損失は 0.999 となった. 学習中の精度は高かったが, 検証用データとテスト用データで共に高い損失を示したため, 過学習が起こったと考えられる. また 2 種類のインプラントを学習させた結果, 5 パターンでテスト用データに対する正答率が 92.2%以上, 損失が 0.236 以下と非常に高い精度を示した. しかしインターナルインプラントとレーザーロックテープの学習では, テスト用データに対する正答率が 57.9%, 損失が 3.029 になり, 他のパターンと比べて低い精度を示した. 低い精度だった原因として, 解像度の悪さが考えられる. 本研究では, 開発環境の都合上, 解像度に制限があったため, 鮮明な特徴を学習することができなかった. この原因の解決策として開発環境の向上が挙げられる. より高性能な環境で学習することで, 鮮明な特徴を捉えることができ, 人間が見分けることができないインプラントでも識別できるのではないだろうか. また本研究では多くの種類のデータを集めることができず, 4 種類のデータを使った学習であった. 歯科の現場で使われているインプラントの種類は更に多く, 実用化を目指すには更に多くのデータを用いて学習しなければならない.

5. まとめ

本研究では, 深層学習の畳み込みニューラルネットワークを用いて, インプラントのレントゲン画像から種類の識別を目指した. 実験は 4 種類の識別と 2 種類の識別に分けて, どのインプラント同士が識別できるかを検討した. 学習データには文献[2]のデータを用いた. 文献[2]から計 11 枚のデータが切り出し, 上下反転・回転・拡大・縮小などの前処理を施し, 計 24,794 枚に水増しした. 学習させた結果, 4 種類の正答率は 61.7%を示した. 2 種類の正答率は 5 パターンで 92.2%以上を示した. しかし 1 つのパターンの正答率が 57.9%を示した. 4 種類の正答率を下げた原因はこのパターンではないかと考えられる. 本研究を歯科で実用化するには課題があり, その 1 つとしてデータ数が挙げられる. 本研究の学習に用いたデータは少なく, 実用化するにあたって多くの種類の識別が求められる. 歯科で扱うインプラントの種類は数十種類で, 本研究と比べて非常に多い. 数十種類のデータを識別するには, 多くのパラメータが必要になり開発環境の向上が求められる.

謝辞

本研究は, 九州大学応用力学研究所の共同利用研究の助成 (2020, 2021) を受けたものです.

参考文献

- [1] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, 『Dropout: A Simple Way to Prevent Neural Networks from Overfitting』, Journal of Machine Learning Research 15 (2014) 1929-1958 Submitted 11/13; Published 6/14.
- [2] 築瀬武史, 竹島明道, 栗山壮一, 大橋功 (2017), 『続・このインプラントなに?他医院で治療されたインプラントへの対応ガイド』, 医歯薬出版株式会社.
- [3] Francois Chollet (2017), 『Deep Learning with Python』, Manning Publications. (=2018, 株式会社クイープ訳『Python と Keras によるディープラーニング』, 株式会社 マイナビ出版).
- [4] 足立悠 (2021), 『機械学習のための「前処理」入門』, 株式会社リックテレコム.
- [5] 斎藤康毅 (2016), 『ゼロから作る Deep Learning —Python で学ぶディープラーニングの理論と実装』, 株式会社トップスタジオ.