

近代書籍のための フォントを自動生成するニューラルネットの構造

竹本有紀^{†1} 石川由羽^{†2} 高田雅美^{†1} 城和貴^{†1}

概要：近代書籍の文字認識に特化した多フォント活字認識手法には膨大な学習データが必要である。しかし、従来の手動による文字画像の収集には限界がある。そこで、Deep Learning を用いて近代書籍の特定フォントの文字画像を自動生成する手法を提案した。フォントを自動生成する NN の精度向上のためには、NN の最適な構造を見つける必要がある。本稿では、様々なパラメータや構造を組み合わせたフォントの自動生成実験を行い、生成される文字画像の精度が高くなる組み合わせを発見する。

キーワード：フォント生成, Deep Learning, ニューラルネットワーク, 近代書籍

Structure of Neural Network Automatically Generating Font for Early-Modern Japanese Printed Books

YUKI TAKEMOTO^{†1} YU ISHIKAWA^{†2} MASAMI TAKATA^{†1}
KAZUKI JOE^{†1}

1. はじめに

パソコンやスマートフォンの普及に伴い、インターネットは日常生活を豊かにする存在となっている。天気やニュースは、インターネットで検索すれば世界中の情報が瞬時に得られる。ネット通販によって、店舗に直接足を運ばなくても様々な商品を購入できる。これらの利点は、インターネットに接続することができれば、場所や時間帯を選ばず、誰もが利用可能な点である。国立国会図書館[1]は、この利点を生かしたサービスをインターネット上で提供している。国立国会図書館デジタルコレクション[2]は、国立国会図書館の蔵書の画像データを閲覧することができる Web サービスである。このサービスでは、図書館で閲覧が困難な貴重な書籍も公開されている。近代書籍と呼ばれる、明治から昭和初期に刊行された書籍もその1つである。しかしながら、書籍の画像データの閲覧には欠点がある。それは、文書内容のテキスト検索ができない点である。そのため、必要な情報が記載されている箇所を探し出すには、書籍に1ページずつ目を通さなければならない。そこで、サービスの利便性向上のために書籍の画像データの早急なテキスト化が求められている。

画像データをテキスト化する手段としては、画像データから1つずつ文字を読み取り、手作業でコンピュータに文字を入力する方法がある。しかしながら、インターネット上で公開されている書籍の数は膨大であり、手作業による

テキスト化には多大なコストがかかる。そこで、コンピュータによって自動で画像データをテキストデータに変換するのが望ましい。一般的な書籍は、光学文字認識 (Optical Character Recognition, OCR) を用いて自動でテキスト化できる。ところが、近代書籍は市販の OCR ソフトでは正確な認識ができない。そこで、近代書籍に特化した認識手法である多フォント活字認識手法[3][4][5]が提案されている。この手法では、文字認識の精度に学習データの数が影響する。正確な文字認識には、近代書籍の文字画像を大量に用意する必要がある。現在、学習データの主要な収集方法は、画像データから手動による文字画像の切り出しである。しかし、手動による収集には限界がある。よって、効率的な収集方法が必要である。

著者は新たな文字画像の収集方法として、Deep Learning[6]を用いて近代書籍の特定の出版者・出版年代のフォントの文字画像を自動生成する手法[7]を提案した。この手法では、それぞれ4つの畳み込み層と逆畳み込み層から成る Neural Network (NN) を構築した。この NN を用いたフォントの自動生成実験では、ゴシック体の文字画像から、明治中期に駿々堂で用いられたフォントの文字画像の生成を試みた。その結果、生成された文字画像は、多フォント活字認識手法の学習データとして利用可能であることが示されている。しかしながら、近代書籍の文字画像との画素値の平均一致率は、学習に用いなかった文字画像から自動生成した場合が 73.69%であった。Deep Learning では、NN の構造やパラメータの値が生成画像の精度に大きく影響する。そこで、本稿では、近代書籍のフォント生成に適した NN の構造を見つけるための実験を行い、自動生

^{†1} 奈良女子大学
Nara Women's University

^{†2} 滋賀大学
Shiga University

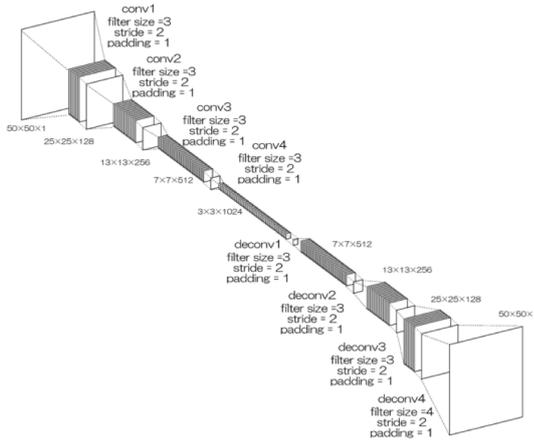


図 1 フォントを自動生成する NN の構造

成される文字画像の精度向上を目指す。

本稿の構成を示す。まず、2 章で近代書籍のフォントを自動生成する NN について紹介する。3 章で近代書籍のフォント自動生成に適した NN の構造を見つけるための実験の方法を説明し、4 章では、実験から得られた最も適した構造の NN でフォント生成を行なった結果を示す。

2. Deep Learning を用いたフォントの自動生成手法

2.1 アルゴリズム

本章では、著者が提案した、Deep Learning を用いて近代書籍の特定の出版者・出版年代のフォントを自動生成する手法について説明する。この手法では、文字画像に文字固有の特徴とフォント固有の特徴があると考え、入力された文字画像の文字固有の特徴を保持しながらフォント固有の特徴のみの変換を目標としている。文字固有の特徴とは、文字を構成する点の位置や線の形状などを表す。フォント固有の特徴とは、点や線の両端の形状、線の太さなどを表す。構築した NN に文字画像を入力すると、近代書籍の特定の出版者・出版年代のフォントに変換された文字画像が出力される。

近代書籍の特定の出版者・出版年代のフォントを自動生成する手順について説明する。まず、学習に用いるデータセットを読み込む。データセットは、フォント変換の元となる現代フォントの文字画像と、特定の出版者・出版年代の近代書籍から切り出した文字画像である。読み込んだデータセットは、訓練データとテストデータに分ける。次に、このデータセットとフォントを自動生成する NN を用いて入力画像のフォントを変換した文字画像を自動生成する。文字画像を生成した後は、訓練データから自動生成された文字画像と近代書籍の文字画像との誤差を算出し、NN に逆伝搬する。各層のニューロンにおいて、計算した誤差を用いてパラメータを更新する。さらに、テストデータから

表 1 訓練データを用いたフォントの自動生成結果

ゴシック体の文字画像	近代書籍から切り出した文字画像	自動生成された文字画像
遙	遙	遙
光	光	光
禮	禮	禮

表 2 テストデータを用いたフォントの自動生成結果

ゴシック体の文字画像	近代書籍から切り出した文字画像	自動生成された文字画像
紀	紀	紀
公	公	公
嬉	嬉	嬉

自動生成された文字画像と近代書籍の文字画像との誤差を求め、NN による生成画像の精度を確認する。これらの工程を一定回数繰り返したら、学習を終了する。学習終了後、テストデータから求めた誤差が最小となるパラメータを持つ NN が、近代書籍のフォントの自動生成に最も適した NN となる。この学習済みモデルを用いて、フォントの自動生成を行うことができる。学習したフォントに変換したい文字の画像を入力すると、フォントが変換された文字画像が生成される。

2.2 フォントを自動生成する NN

フォントを自動生成する NN について説明する。NN に現代フォントの文字画像を入力すると、フォントが変換された文字画像が出力される。フォントを自動生成する NN の構造を図 1 に示す。NN はそれぞれ 4 つの畳み込み層と逆畳み込み層によって構成される。フィルタサイズは 3x3 である。ただし、画像サイズを入力と出力で統一するため、最後の逆畳み込み層のみ 4x4 とする。各層の出力チャンネル数は、外側から順に 128、256、512、1024 である。padding の幅は 1、stride は 2 に設定する。stride を 2 にすることで、プーリングとアンプーリングの役割を担う。活性化関数には $f(x) = \max(0, x)$ で表される ReLU 関数を用いる。4 つの畳み込み層には、過学習を防ぐため dropout 関数[8]を適用する。dropout 率は 0.7 に設定する。

2.3 フォントの自動生成実験

この手法による NN を用いた実験では、1297 組のデータセットを用いてフォントの自動生成を行った。近代書籍の文字画像は、全て駿々堂で明治中期に出版された書籍から切り出したものである。データセットのうち 1200 組を訓練データ、残りの 97 組をテストデータに用いた。表 1 に訓練データを用いてフォントを自動生成した結果を示す。表 2 はテストデータを用いてフォントを自動生成した結果である。表の左の列はゴシック体の文字画像、中央の列は近代書籍の文字画像、右の列は自動生成された文字画像である。3 種類の文字画像を比較すると、入力画像に用い

表 3 フォント変換に失敗した文字画像の例

近代書籍から 切り出した 文字画像	自動生成された 文字画像
十 字 雲	十 字 雲
十 字 雲	十 字 雲

たゴシック体のフォントが変換され、近代書籍に近いフォントの文字画像が生成されていることが分かる。自動生成された文字画像と近代書籍の文字画像において画素値の平均一致率を算出すると、訓練データから自動生成した場合が 99.79%、テストデータから自動生成した場合が 73.69%であった。訓練データに比べ、テストデータの方が平均一致率が大幅に低い。そこで、テストデータから自動生成した文字画像と近代書籍の文字画像で PDC 特徴を比較した。PDC 特徴は多フォント活字認識手法で文字の特徴抽出に用いられる。そのため、自動生成した文字画像と近代書籍の文字画像の PDC 特徴が類似していれば、自動生成された文字画像は多フォント活字認識手法の学習データに利用可能であるといえる。比較の結果、テストデータから自動生成した文字画像と近代書籍の文字画像の PDC 特徴は類似していた。このことから、フォントを自動生成する NN によって、近代書籍の文字画像のフォントを再現した文字画像が自動生成され、多フォント活字認識手法の学習データを補完することができると示した。

フォントを自動生成する NN は、入力された文字画像のフォントを変換し、学習データに用いた近代書籍のフォントの再現に成功した。しかし、学習に用いていない未知の文字に対しては、フォントの変換が不十分で文字の構成が崩れてしまう場合もある。フォント変換に失敗した文字画像の例を表 3 に示す。表中の自動生成された文字画像は、フォントが変換できなただけでなく、文字を構成する特徴も失われている。より正確なフォント変換を行うためには、フォントを自動生成する NN の構造を改良する必要があると考えられる。そこで、フォントの自動生成に適した NN の構造を見つけるために、フォントの自動生成実験を行う。その詳細は 3 章で述べる。

3. フォントの自動生成実験

NN の構造において、自動生成された文字画像の精度に影響する要素として次の 5 つが考えられる。

- I. 畳み込み層の層数
- II. 各層の出力チャンネル数
- III. フィルタサイズ
- IV. プーリングの種類
- V. プーリングの回数

畳み込み層の層数が多いほど、文字画像の持つ特徴を深

く学習できる。出力チャンネル数が多いほど、文字画像から抽出できる特徴の数が増える。フィルタサイズが大きいほど、文字画像から大きな特徴を抽出できる。これらのパラメータが大きくなるほど、訓練データからより多くの特徴を学習できるようになると考えられる。しかし、訓練データから特徴を学習しすぎると、学習に用いない未知データに対する精度は下がる。そこで、テストデータに対して精度が高いパラメータの組み合わせを実験によって見つける。

プーリングは、文字画像内における位置のずれを補正するために行う。プーリングを繰り返すと、画像サイズが縮小し、文字画像の持つ細部の特徴が失われていく。そのため、適切な回数を見つける必要がある。また、その種類によってプーリング層から出力される画像の画素値が変化する。そのため、プーリングの種類によっては、フォント変換に適するものとそうでないものがあると考えられる。

フォントの自動生成に適した NN の構造を見つけるため、5 つの要素のパラメータを変化させて実験を行う。この実験で試行した組み合わせを以下に示す。

- I. 畳み込み層の層数
2 層, 3 層, 4 層
- II. 1 層目の出力チャンネル数
8, 16, 32, 64, 128
- III. フィルタサイズ
3x3, 5x5, 7x7, ...
- IV. プーリングの種類
max pooling, average pooling
- V. プーリングの回数
全層, 外側 1 層, 外側 2 層, ...

畳み込み層の層数は、2 層, 3 層, 4 層の場合に分けてそれぞれフォントの自動生成を行う。出力チャンネル数は 1 層目の最小値を 8, 最大値を 128 とする。2 層目以降は、1 層ごとに 2 倍にする。フィルタサイズは 3x3 から 2 ずつ大きくしていく。プーリングの種類は、max pooling, average pooling とする。回数は、畳み込み層が 2 層の場合は全層で行う。3 層の場合は全層, 外側 2 層, 外側 1 層とする。4 層の場合は全層, 外側 3 層, 外側 2 層, 外側 1 層とする。

フォントの自動生成実験について説明する。学習に用いるデータセットにおいて、近代書籍の文字画像は、駿々堂で明治中期に出版された書籍から切り出したものとする。データセットは、ゴシック体と近代書籍の文字画像それぞれ 1 枚、計 2 枚を 1 組とし、500 組用意する。450 組を訓練データ、残りの 50 組をテストデータとして学習する。画像サイズは 64x64 に統一する。学習回数は 5000 回、ミニバッチサイズは 50 とする。学習率は、過学習を抑制し、テストデータから自動生成された画像と駿々堂明治中期フォントの文字画像の平均画素値一致率が上がるように適宜調整す

る。

4. 最適な構造の NN によるフォントの自動生成

フォントの生成実験の結果から画素値の平均一致率が高くなる構造とパラメータを選び、1297組のデータセットを用いてフォントの自動生成を行った。その結果、畳み込み層4層、出力チャンネル数(64, 128, 256, 512)、フィルタサイズ7x7、max poolingを外側2層に適用した場合、テストデータに対する画素値の平均一致率が最も高かった。このとき、訓練データに対する画素値の平均一致率は76.09%、テストデータに対する画素値の平均一致率は75.05%であった。このNNの畳み込み層のうち、3層目と4層目にdropout関数を適用すると、dropout率0.7の時、訓練データに対する画素値の平均一致率は76.48%、テストデータに対する画素値の平均一致率は75.71%となった。テストデータに対する画素値の平均一致率はわずかに高くなったが、般化率の向上には至らなかった。フォントの生成実験において、最も再現精度の高いNNでは、訓練データに対する画素値の平均一致率は100%、テストデータに対する画素値の平均一致率は78.41%であった。学習データの数を増やした結果、1297組のデータセットを用いると、フォントの生成実験の時に比べて画素値の平均一致率が低くなっている。これは、文字の種類が増えたことにより、学習しなければならない特徴が増え、NNが十分に学習できなかったためである。そこで、画像サイズを64x64から32x32に変更したところ、畳み込み層4層、出力チャンネル数(128, 256, 512)、フィルタサイズ5x5、max poolingを外側2層に適用した場合、訓練データに対する画素値の平均一致率は99.32%、テストデータに対する画素値の平均一致率は75.79%となった。新たな構造のNNによるフォント自動生成では、いずれの場合も既存の手法より良い結果を示している。このことから、NNの構造によって生成画像の精度が向上することが分かった。今回は、各畳み込み層のフィルタサイズを統一して実験を行なった。しかしながら、プーリングを行うと画像サイズが変化するため、層ごとにフィルタサイズを変えた方が、より精度が向上すると考えられる。今後はさらに実験を行い、フォント変換の精度向上を目指す。

謝辞 本研究はMEXT科研費JP17H01829の助成を受けたものです。

参考文献

- [1] 国立国会図書館. <http://www.ndl.go.jp>. Accessed 2019-07-01.
- [2] 国立国会図書館デジタルコレクション. <http://dl.ndl.go.jp>. Accessed 2019-07-01.
- [3] Chisato Ishikawa, Naomi Ashida, Yurie Enomoto, Masami Takata, Tsukasa Kimesawa and Kazuki Joe. "Recognition of Multi-Fonts Character in Early-Modern Printed Books". Proceedings of International Conference on Parallel and Distributed Processing

- Techniques and Applications (PDPTA09), Vol. II, pp. 728-734(2009).
- [4] Manami Fukuo, Yurie Enomoto, Naoko Yoshii, Masami Takata, Tsukasa Kimesawa and Kazuki Joe. "Evaluation of the SVM based Multi-Fonts Kanji Character Recognition Method for Early-Modern Japanese Printed Books". Proceedings of The 2011, International Conference on Parallel and Distributed Processing Technologies and Applications (PDPTA2011), Vol. II, pp. 727-732(2011).
 - [5] 粟津妙華, 上坂和美, 高田雅美, 城和貴: 近代書籍を対象とした多フォント活字認識手法, 情報処理学会論文誌. 数理モデル化と応用(TOM), Vol. 9(2), pp. 33-40(2016).
 - [6] 岡谷貴之, 齋藤真樹. ディープラーニング. 情報処理学会研究報告, Vol.2013-CVIM-185, No.19, pp.1-6, 2013.
 - [7] Yuki Takemoto, Yu Ishikawa, Masami Takata, Kazuki Joe. : Automatic Font Generation for Early-Modern Japanese Printed Books, Proceedings of The 2018 International Conference on Parallel and Distributed Processing Technologies and Applications (PDPTA2018), Vol. I, pp. 326-332(2018).
 - [8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". Journal of Machine Learning Research 15 (2014) 1929-1958.