

# 機械学習を用いた筆跡鑑定

大井 健太<sup>1,\*</sup> 杉田 誠<sup>1,2,†1</sup>

**概要:** 筆跡鑑定の結果は裁判での重要な証拠となるため経験と知識を持つ専門家による鑑定が重要となる。しかし、現在の日本では筆跡鑑定人を名乗るための資格は必要とされていない。そのため筆跡鑑定人の信頼性に鑑定結果が大きく左右されてしまうという状況が起こっている。筆跡鑑定に有用な特徴の抽出を専門家の知識を使用せずに行うことができればよいと考え、機械学習を用いて筆者識別実験を行い、抽出された特徴が筆跡鑑定に利用できないか調査した。本稿では、収集された筆跡を画像と捉え既に画像認識について有効性が確認されている畳み込みニューラルネットワーク(CNN)を用いて筆者識別実験を行った。CNNを用いたモデルによる実験で筆跡鑑定人の信頼性に左右されない特徴抽出が可能ということが分かった。

**キーワード:** 筆跡鑑定 機械学習 ニューラルネットワーク

## Handwriting Analysis Using Machine Learning

Kenta Ooi<sup>1,†1</sup> Makoto Sugita<sup>1</sup>

**Abstract:** Since the results of handwriting appraisal are important evidence in court judgment, it is important for experts with experience and knowledge to expertise. However, in Japan it is not necessary to qualify for identifying a handwriting expert. As a result, the situation that the appraisal result is largely influenced by the reliability of the handwriting expert witness has occurred. I thought that it would be good if we could extract features useful for handwriting experts without using expert knowledge. Therefore, we conducted an author identification experiment using machine learning and investigated whether extracted features can be used for handwriting expertise. In this paper, we conducted a writer identification experiment using a convolution neural network (CNN), which has already gathered handwriting as an image and has already been confirmed valid for image recognition. Experiments using CNN model showed that feature extraction which is not influenced by reliability of handwriting experts is possible.

**Keywords:** Handwriting, Machine Learning, Neural network

### 1. はじめに

#### 1.1 研究背景

筆跡に関する研究には大きく分けて5つあり、その中の法科学の分野で筆跡鑑定(筆者識別)が研究されている[1]。筆跡鑑定とは、「複数の筆跡」を「比較検討」しそれを書いた筆者が同一人によるものか否かを識別する方法である。筆跡鑑定は刑事事件において契約書や領収書などの書類が違法に作成された可能性が考えられる場合、それらの書類がどのような筆者により書かれたものかを調べるために行われる[2]。筆跡鑑定では、本人筆跡と疑問筆跡についてそれぞれ筆者固有の書き癖を抽出し、その癖同士を比較し筆者識別を行っている。しかし、現在の日本では筆跡鑑定をする筆跡鑑定人になるために資格が必要とされていない。そのため、鑑定結果が鑑定人の信頼性に左右されてしまうという状況にある。そこで、画像認識について既に成果がある畳み込みニューラルネットワークを用いて筆者固有の特徴を抽出する手法を提案する。

#### 1.2 先行研究

[3]においては、機械学習を用いたパターン認識による筆者識別を提案しており、濃度ヒストグラム、画素値、文字の占有率を特徴値としてニューラルネットワークを用いている。

[2]においては、条件付き Auto Encoder による書き癖抽出手法の提案しており、画像から畳み込みを利用した条件付き Auto Encoder を用いている。

#### 1.3 研究目的

筆跡鑑定(筆者識別)に有用な特徴の抽出を筆跡鑑定の専門家の知識を使用せずに行う。

## 2. 概要および実験環境

#### 2.1 概要

2人分の筆跡( $\alpha$ ,  $\beta$ )をそれぞれ40個ずつ(学習用データ:20個, テスト用データ:20個)用いて2種類(A, B)の筆者識別実験を行いその結果を検討した。Aでは文字列に対して、Bでは文字に対して筆者識別実験を行った。

<sup>1</sup> 公立諏訪東京理科大学工学部情報応用工学科  
Department of Applied Information Engineering, Faculty of Engineering,  
Suwa University of Science  
<sup>†1</sup> 現在, マリモ電子工業株式会社

## 2.2 実験環境

開発・実験に用いた環境を表1に示す。

表1 開発・実験環境

	種類	名称	
ハードウェア	CPU	Intel®, Core™i7-8700K 3.7GHz	
	GPU	NVIDIA, GeForce GTX 1080Ti	
	イメージスキャナ	DocuScan C4250	
ソフトウェア	OS	Windows10 Pro	
	言語	Python3.5.6	
	ライブラリ		tensorflow-gpu 1.11.0
			Keras 2.2.4
			opencv-python 3.4.3.18
		numpy 1.15.2	

## 2.3 筆跡収集

筆者識別実験で使用する筆跡データは、既定の用紙の枠内にボールペンを用いて筆記されたものを収集した。学習用データ、テスト用データ共に同じフォーマットで収集し収集日を別日とした。筆記する文字は漢字で「諏訪理科」とし、それを一人につき40個集めた。

筆跡収集用紙の一部を図1に示す。

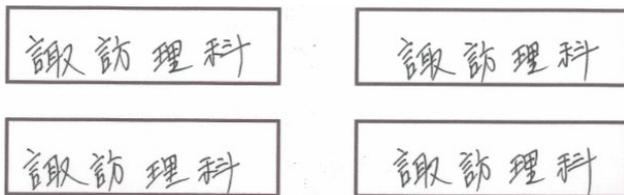


図1 筆跡収集用紙

## 3. 提案方式 A

### 3.1 実験手順

フローチャートを用いて実験手順を図2に示す。



図2 実験手順

正規化工程では学習工程の入力に合うよう各種処理を行っ

た。学習工程では筆跡画像より筆者個人の特徴を抽出したモデルを作成した。テスト工程では、テスト用データを用いて学習済みモデルに入力し、モデルの精度を検証した。表2に正規化工程のそれぞれの処理過程に対する処理内容を示す

表2 正規化工程の処理内容

処理工程	処理内容
デジタルデータ化	既定の用紙を用いて収集した筆跡をイメージスキャナを用いてデジタル画像化した。解像度を600dpiとした。
枠内切り出し 2値化	デジタル画像化された既定の用紙の枠内を切り出した。大津の2値化を用いて切り出された画像を2値化した。
ノイズ除去	紙やイメージスキャナに付着した塵やゴミをWindowsに標準搭載されているペイント(.Ink)を用いて白で塗りつぶした。
筆跡領域取得	枠内で切り出された画像を筆跡を含む最小の矩形領域で切り出した。
リサイズ	100×387にリサイズした。
水増し	学習用データをアフィン変換を用いて1160枚に水増しした。

表3に正規化工程の中の水増し工程のパラメータについて示す。学習用データの水増し時にはアフィン変換を用いている。回転の変換では、筆跡画像を-15度から15度の範囲で、水平移動の変換では、筆跡画像の横の長さ387px\*20%まで、垂直移動の変換では、筆跡画像の縦の長さ100px\*20%までの間で無作為に行った。

表3 水増し工程のパラメータ

変換の種類	パラメータ
回転	-15度~15度
水平移動	20%
垂直移動	20%

図3に学習工程で用いたモデルを示す。

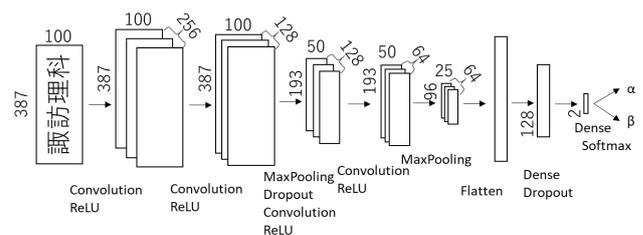


図3 学習に用いたモデル

提案方式Aの実験で用いたモデル図3では、1層目の畳み込み層の後の特徴マップを256個、2層目の畳み込み層の後の特徴マップを128個、3層目の畳み込み層の後の特徴マップを128個、4層目の畳み込み層の後の特徴マップを64個用いた。その後、3次元配列を1次元配列に平滑化し、128個のノードを持つ全結合層を繋げた。その後、2クラス分類なので2個のノードを持つ全結合層を繋げ、Softmax関数を用いて出力を確率として出力した。出力層以外では、活性化関数にReLUを用いた。用いた活性化関数について式(1)と(2)に示す。

$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

式(1) ReLU 関数

$$y = \frac{e^{(a_k)}}{\sum_{i=0}^n e^{(a_i)}}$$

式(2) Softmax 関数

表 4 に図 3 のモデルの学習時に用いたハイパーパラメータを示す。最適化アルゴリズムに、Adam を用いた。その時の学習率は $(1 \times e)^{-4}$ とした。バッチサイズは 40 とし、エポックは 14 回で打ち切った。損失関数に Binary Cross Entropy を用いた。Adam について式(3)、Binary Cross Entropy について式(4)に示す。

表 4 学習時のハイパーパラメータ

パラメータ	設定値
最適化アルゴリズム	Adam(lr=1e-4)
バッチサイズ	40
エポック	14
損失関数	BinaryCrossEntropy

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla E(w^t)$$

$$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \nabla E(w^t)^2$$

$$\hat{m} = \frac{m_{t+1}}{1 - \beta_1^t}$$

$$\hat{v} = \frac{v_{t+1}}{1 - \beta_2^t}$$

$$w^{t+1} = w^t - \alpha \frac{\hat{m}}{\sqrt{\hat{v} + \epsilon}}$$

式(3) Adam

$$E(w) = - \sum_n^N (d_n \log y_n + (1 - d_n) \log(1 - y_n))$$

式(4) Binary Cross Entropy

表 5 にモデルのハイパーパラメータを示す。1 層目と 2 層目ではサイズ 5×5 のフィルタ、3 層目と 4 層目ではサイズ 3×3 のフィルタを適用した。プーリング層では、2×2 の Max Pooling を行った。Dropout 層では 50% の不活性化を行った。

表 5 モデルのハイパーパラメータ

レイヤータイプ	フィルタサイズ	プーリングサイズ	不活性化率(%)
1 Convolution	5×5		
2 Convolution	5×5		
3 Convolution	3×3		
4 Convolution	3×3		
1 MaxPooling		2×2	
2 MaxPooling		2×2	
1 Dropout			50
2 Dropout			50

### 3.2 学習結果

学習工程の学習経過の様子を図 4 に示す。

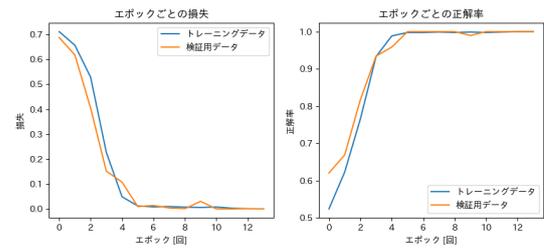


図 4 学習経過

図 4 のエポックごとの損失のグラフより損失関数の値が収束していることがわかる。そのため学習をエポック 14 回で打ち切った。エポックごとの正解率のグラフよりトレーニングデータ、検証用データ共に 100% 近い正解率を出していることが分かる。现阶段では、過学習も起こしていないのでモデル作成を成功したといえる。

### 3.3 テスト結果

テスト用データを入力したときのテスト結果を表 6、表 7 に示す。ともに各ラベルに対する正解率を高く示していることが分かる。よってモデルは筆者識別をできているといえる。

表 6  $\alpha$  の筆跡画像データ入力時

表 7  $\beta$  の筆跡画像入力時

$\alpha$		$\beta$		
1.0000000000	0.0000003065	0.0001340619	0.9998658895	
1.0000000000	0.0000000940	0.00000520172	0.99999475479	
1.0000000000	0.0000003294	0.00002203090	0.99997794628	
1.0000000000	0.0000003614	0.00000517631	0.99999487400	
0.9999988079	0.0000006172	0.00000490981	0.99999511242	
0.9999904633	0.00000090642	0.00000253135	0.99999749660	
1.0000000000	0.0000000597	0.00000235337	0.99999761581	
1.0000000000	0.0000003385	0.00000038189	0.9999964237	
0.9999976158	0.00000020931	0.00001255188	0.99998748302	
1.0000000000	0.00000005680	0.00000454117	0.99999547005	
0.9999940395	0.00000057896	0.00000174087	0.99999821186	
0.9999976158	0.00000019290	0.00000405761	0.99999594688	
0.99999535084	0.00000461159	0.00007577993	0.99992418289	
0.9999976158	0.00000022575	0.00001037634	0.99998962879	
0.9999988079	0.00000006723	0.00000820594	0.99999177456	
1.0000000000	0.00000005562	0.00000237090	0.99999761581	
1.0000000000	0.00000004095	0.00000718994	0.99999284744	
1.0000000000	0.00000000080	0.00004361862	0.99995636940	
1.0000000000	0.00000004346	0.00000302461	0.99999701977	
0.9999988079	0.00000006177	0.00000581580	0.99999415874	
平均	0.9999963641	0.00000036311	0.00001779596	0.99998220205

### 3.4 可視化

図 5\_1, 図 5\_2, 図 6\_1, 図 6\_2 に Grad-CAM を用いてテスト用データに対して可視化を行った結果を示す。

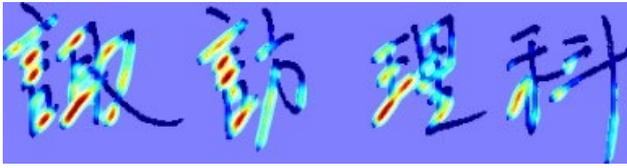


図 5\_1  $\alpha$  のテストデータ

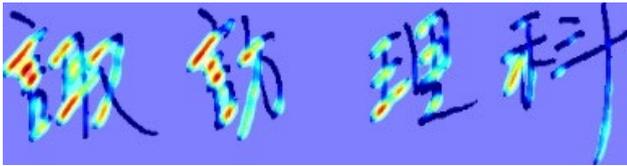


図 5\_2  $\alpha$  のテストデータ

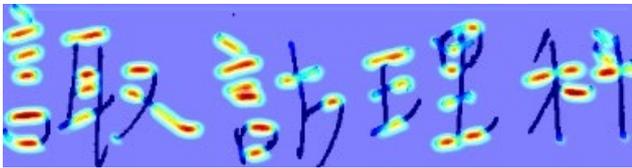


図 6\_1  $\beta$  のテストデータ

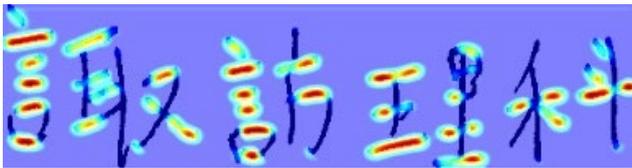


図 6\_2  $\beta$  のテストデータ

上の図より特徴ととらえている部分の可視化ができていることが分かる。同じような箇所を特徴ととらえていることが分かる。

## 4. 提案方式 B

### 4.1 実験手順

フローチャートを用いて実験手順を図 7 に示す。

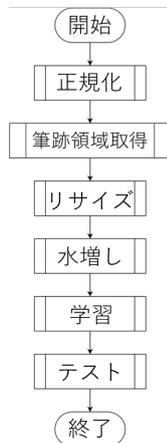


図 7 実験手順

正規化工程では学習工程の入力に合うよう各種処理を行った。学習工程では筆跡画像より筆者個人の特徴を抽出したモデルを作成している。テスト工程では、テスト用データを用いて学習済みモデルに入力し、モデルの精度を検証した。

表 8 に正規化工程のそれぞれの処理過程に対する処理内容を示す。文字列として集められた筆記データを使用し実験を行った。そのため筆跡領域取得の工程では「諏訪理科」の中の「諏」の文字を実験対象文字として切り取った。その後のリサイズ領域では、切り出された画像を白塗りのブランク画像の中心に配置することによりすべての文字の大きさの差異を無くした。水増し工程では、表 3 のパラメータを用いて水増しを行った。

表 8 正規化工程の処理内容

処理工程	処理内容
デジタルデータ化	既定の用紙を用いて収集した筆跡をイメージスキャナを用いてデジタル画像化した。解像度を600dpiとした。
枠内切り出し	デジタル画像化された既定の用紙の枠内を切り出した。
2値化	大津の2値化を用いて切り出された画像を2値化した。
ノイズ除去	紙やイメージスキャナに付着した塵やゴミをWindowsに標準搭載されているペイント.Inkを用いて白で塗りつぶした。
筆跡領域取得	文字に対する実験として「諏」を切り出した。切り出された画像を筆跡を含む最小の矩形領域で切り出した。
リサイズ	190×190の白で塗りつぶされたブランク画像の中心に筆跡を配置した。
水増し	学習用データをアフィン変換を用いて1160枚に水増しした。

図 8 に実験で用いたモデルを示す。

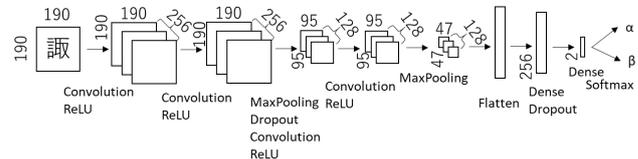


図 8 学習に用いたモデル

提案方式 B の実験で用いたモデル図 7 では、1 層目の畳み込み層の後の特徴マップを 256 個、2 層目の畳み込み層の後の特徴マップを 256 個、3 層目の畳み込み層の後の特徴マップを 128 個、4 層目の畳み込み層の後の特徴マップを 128 個用いた。その後、3 次元配列を 1 次元配列に平滑化し、256 個のノードを持つ全結合層を繋げた。その後、2 クラス分類なので 2 個のノードを持つ全結合層を繋げ、Softmax 関数を用いて出力を確率として出力した。出力層以外では、活性化関数に ReLU を用いた。用いた活性化関数について式(1)と(2)に示す。

表 9 に図 8 のモデルの学習時に用いたハイパーパラメータを示す。最適化アルゴリズムに、Adam を用いた。その時の学習率は $(1 \times e)^{-4}$ とした。バッチサイズは 40 とし、エポックは 9 回で打ち切った。損失関数に Binary Cross Entropy を用いた。Adam について式(3)、Binary Cross Entropy について式(4)に示す。

表 9 学習時のハイパーパラメータ

パラメータ	設定値
最適化アルゴリズム	Adam(lr=1e-4)
バッチサイズ	40
エポック	9
損失関数	BinaryCrossEntropy

表 5 にモデルのハイパーパラメータを示す。1 層目と 2 層目ではサイズ 5×5 のフィルタ、3 層目と 4 層目ではサイズ 3×3 のフィルタを適用した。プーリング層では、2×2 の Max Pooling を行った。Dropout 層では 50% の不活性化を行った。

#### 4.2 学習結果

学習工程の学習経過の様子を図 9 に示す。

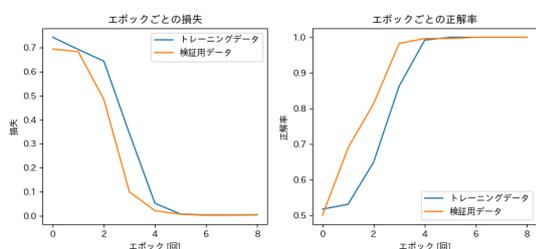


図 9 学習経過

図 4 のエポックごとの損失のグラフより損失関数の値が収束していることがわかる。そのため学習をエポック 9 回で打ち切った。エポックごとの正解率のグラフよりトレーニングデータ、検証用データ共に 100% 近い正解率を出していることが分かる。現段階では、過学習も起こしていないのでモデル作成を成功したといえる。

#### 4.3 テスト結果

テスト用データを入力したときのテスト結果を表 10、表 11 に示す。ともに各ラベルに対する正解率を高く示していることが分かる。よってモデルは筆者識別をできているといえる。

表 10  $\alpha$  の筆跡画像データ入力時

$\alpha$	$\beta$
0.99995875359	0.00004119149
0.99995028973	0.00004970017
0.99961346388	0.00038651211
0.99997794628	0.00002206803
0.99997520447	0.00002474427
0.99965429306	0.00034571925
0.99999034405	0.00000969025
0.99997210503	0.00002789163
0.99914348125	0.00085654680
0.99983036518	0.00016955519
0.99994003773	0.00005992511
0.99904233217	0.00095769722
0.99995768070	0.00004236223
0.99995684624	0.00004316926
0.99999487400	0.00000508686
0.99852633476	0.00147366151
0.99999594688	0.00000400027
0.99985837936	0.00014160728
0.99420845509	0.00579153886
0.99991941452	0.00008057421

平均 0.99947332740 0.00052666210

表 11  $\beta$  の筆跡画像データ入力時

$\alpha$	$\beta$
0.00000115525	0.99999880791
0.00000090550	0.99999904633
0.00000385434	0.99999618530
0.00000284157	0.99999713898
0.00000069627	0.99999928474
0.00000017425	0.99999988079
0.00000021487	0.99999976158
0.00002042229	0.99997961521
0.00000649312	0.99999356270
0.00000003437	1.00000000000
0.00007399942	0.99992597103
0.00000210791	0.99999785423
0.00000314603	0.99999690056
0.00000985018	0.99999010563
0.00000076258	0.99999928474
0.00000040158	0.99999964237
0.00040976977	0.99959021807
0.00000095070	0.99999904633
0.00000058165	0.99999940395
0.00000226376	0.99999773502

平均 0.00002703127 0.99997297227

#### 4.4 可視化

図 10\_1, 図 10\_2, 図 11\_1, 図 11\_2 に Grad-CAM を用いてテスト用データに対して可視化を行った結果を示す。



図 10\_1  $\alpha$  のテストデータ



図 10\_2  $\alpha$  のテストデータ

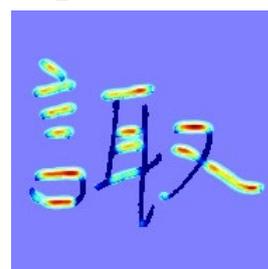


図 11\_1  $\beta$  のテストデータ

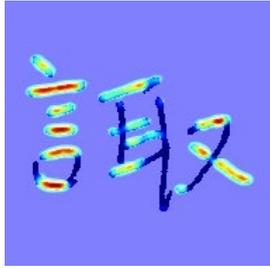


図 11\_2  $\beta$  のテストデータ

上の図より特徴ととらえている部分の可視化ができて  
いることが分かる。同じような箇所並びを特徴ととらえ  
ていることが分かる。

## 5. まとめ

筆者固有の特徴を抽出する際に、畳み込みニューラルネ  
ットワークが有効であることが分かった。

先行研究[3]に対して、畳み込みニューラルネットワー  
クを使用することで画素ごとのつながりを保持すること  
ができていたので、同様のデータを用いて実験を行った  
際に精度がよくなる可能性がある。

オフラインの筆者識別では得られる特徴値が少ないとい  
う欠点があげられる。その欠点を解決し、それらの特徴  
値を用いてパターン認識を行う必要がある。

## 参考文献

- [1] 法科学鑑定研究所 筆跡鑑定 : <http://alfs-inc.com/hituseki/>
- [2] 細江麻梨子, 山田智輝, 加藤邦人, 山本和彦, “条件付き  
AutoEncoder による書き癖抽出手法の提案“, 情報処理学会第  
80 回全国大会
- [3] 高橋真奈茄, 小出洋, “機械学習を用いたパターン認識によ  
る筆者識別”, . 第 57 回 プログラミング・シンポジウム  
2016.1.8-10
- [4] “学習の種類と誤差関数” : [http://bcl.sci.yamaguchi-](http://bcl.sci.yamaguchi-u.ac.jp/~jun/notebook/keras/loss-)  
[u.ac.jp/~jun/notebook/keras/loss-](http://bcl.sci.yamaguchi-u.ac.jp/~jun/notebook/keras/loss-)