

# CNN及びLSTMによる筆跡鑑定とその判断根拠の可視化

齊藤孝介 今井一真 大塚一允

著者 齊藤孝介

群馬県立高崎高等学校

## 要旨

現行の筆跡鑑定が抱える問題を解消すべく筆跡鑑定AIの開発が行われているが、これらのAIには判断根拠が不明瞭という問題がある。本研究では、3種の方法で筆跡鑑定AIを実装するとともに、その判断根拠の可視化方法について考察と検証を行った。筆跡の時系列データを用いた判断根拠の可視化には成功したが、その理由や傾向を人間にとって説明可能な形で示すにはさらなる研究が必要である。

## 1. はじめに

### 1.1 研究背景

筆跡鑑定は裁判の判決を下す根拠にもなりうる必要不可欠なものであるが、いくつかの問題点がある。第一に、日本には現在筆跡鑑定の資格が存在しない。加えてその判断基準も統一されていないため、鑑定結果の信頼性はどの鑑定士が携わったかに大きく左右される。これらを背景に、人工知能によって筆跡鑑定を行うことに期待が寄せられている。関陽子による研究「ニューラルネットワークによる筆跡の分類」<sup>1</sup>では、CNNを用いて10人の筆跡を分類することに成功している。また吉間優理、六井淳による研究「筆跡特徴量余韻を用いたLSTMに基づくオンライン署名認証」<sup>2</sup>では、LSTMを用いて筆跡ストロークにより筆者を識別する手法を提案している。

人工知能による識別もまた別の問題を抱えている。機械学習において基本的な手法の一つである深層学習は学習のプロセスにおいて判断基準そのものを学習するという特長があり、この利便性をもってあらゆる分野への導入が期待されている。その反面、この自律性により人工知能が下した判断の根拠を人間が理解できないことが大きな問題とされており、これを解決すべく近年研究が進められているのが説明可能なAI(XAI)である。

本研究の目的は、筆跡鑑定AIの判断根拠を可視化し、人間にとって説明可能な形で示すことである。

### 1.2 研究概要

CNNやLSTMなど複数の手法を用いて筆跡鑑定AIを実装し、精度を比較した。次に判断根拠の可視化システムを各手法に合わせて実装し、実際に出力してその解釈可能性について考察した。

## 2. 研究方法

### 2.1 データ収集（実施時期：2023年1月、6月）

6人から筆跡データ「hand」を計約650文字分収集した。筆跡データとしては、画像データと、各瞬間におけるペン先の座標を並べた時系列データの二種類を用いた。収集には先述の二種類のデータを同時に保存できる自作のツールを使用した。キャンバスサイズは300px×600pxの横長長方形である。650文字うち64%を学習用、16%を検証用、20%をテスト用データに分割した。時系列データについては0でパディングを行い長さを一定にした。

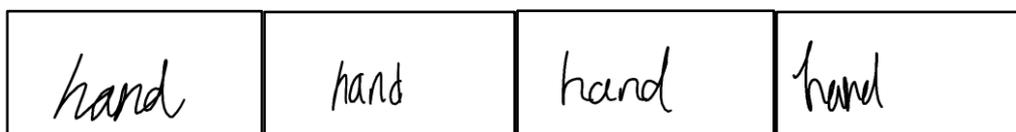


図1. 収集した画像データの例

<sup>1</sup> 関陽子.ニューラルネットワークによる筆跡の分類.日本教育心理学会第61回総会発表論文集.2019,p341

<sup>2</sup> 吉間優理, 六井淳.筆跡特徴量余韻を用いたLSTMに基づくオンライン署名認証.人工知能学会全国大会論文集.2022,vol.36

## 2.2 実装方法A: 画像データを用いたCNNによる実装 (実施時期: 2023年6月)

CNN(畳み込みニューラルネットワーク)は画像の学習に広く利用されている深層学習の一手法である。全ての学習用画像データを学習させて以下のモデルを作成した。

層構成は以下の通りである。まずカーネルサイズ(3, 3)の畳み込み層を2つ重ね、プールサイズ(2, 2)のMax Pooling層をつける。次に同じ様に畳み込み層2つとMax Pooling層1つをつけ、さらに畳み込み層を2つ繋げる。その後平滑化、Dropoutを行い、全結合層を2つ重ねてSoftmax関数により予測確率を出力する。出力層以外の活性化関数には全てReLUを、最適化関数にはAdadelta、損失関数にはCategorical Cross Entropyを用いた。

バッチサイズは8、エポック数は256で学習を行った。このエポック数は精度がおおよそ収束するエポック数を調べたのち決定したものであり、これは後の実装方法B、実装方法Cについても同様である。

### 判断根拠の可視化方法について

Simonyanらによる研究「Deep inside convolutional networks: Visualising image classification models and saliency maps」<sup>3</sup>では、画像データの分類における入力データの影響度を以下の式により計算する手法を提案している。

$$w = \frac{\partial S_c}{\partial I} \Bigg|_{I_0} \quad \begin{array}{l} S_c: \text{the score of the class } c \\ I: \text{the image} \\ w: \text{the importance of the image (pixels) for the class } c \end{array}$$

今回の実験では、筆跡鑑定AIの判断根拠の可視化をこの手法に則り試みた。具体的には、まず入力データについて損失関数を微分し、その値、すなわち勾配の絶対値を重要度と定義する。この重要度が大きいほど判断上対応するピクセルの情報が重要であったと解釈した。そして元の画像データの各ピクセルを重要度が高いほど赤に近く、低いほど紫に近くなるよう着色してこれを判断根拠の可視化された図として出力した。

## 2.3 実装方法B : 時系列データを用いたLSTMによる実装 (実施時期: 2023年6月)

LSTM(Long Short-Term Memory)はRNN(再帰型ニューラルネットワーク)の課題である勾配消失を低減する機械学習の手法であり、自然言語処理をはじめとした時系列データの学習に広く活用されている。全ての学習用時系列データを学習させて以下のモデルを作成した。

層構成は以下のとおりである。第一に、時系列データは0によってパディングされているので入力層でマスキングを行う。次にユニット数30のLSTM層を2つ重ね、Dropoutを行った後Dense層に繋げてSoftmax層により予測確率を出力する。出力層以外の活性化関数には全てReLUを、最適化関数にはAdam、損失関数にはCategorical Cross Entropyを用いた。

学習率は0.01、バッチサイズは16、エポック数は256で学習を行った。

判断根拠については実装方法Aと同様に損失関数の勾配を利用して可視化した。出力の際は、各瞬間のペン先の座標に従って点をプロットしたものを着色し、画像と同じように「hand」の文字が見える形で可視化した。

## 2.4 実装方法C : 時系列データを用いたCNNによる実装 (実施時期: 2023年7月)

CNNは画像認識に使われることが多いが、その特長を別種のデータでも生かすことができる場合がある。この実装方法Cでは、実装方法BにおいてLSTMで学習させた時系列データをCNNで学習させることを試みた。時系列データは二次元上の点の座標を各瞬間について集めたものであるから、形は(データの長さ, 2(x座標とy座標))となり、これをサイズが(データの長さ)×2 の二次元画像データと捉えて学習させた。こちらについても全ての学習用時系列データを学習させて以下のモデルを作成した。

<sup>3</sup> Karen Simonyan, Andrea Vedaldi, Andrew Zisserman.

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv.2014-04-19

層構成は以下の通りである。最初にBatch Normalizationを行い、続けて0についてマスキングを行う。次にカーネルサイズ(10, 2)と(10, 1)の畳み込み層を順に重ね、プールサイズ(3, 2)のMax Pooling層をつける。さらにカーネルサイズ(10, 1)の畳み込み層2つとプールサイズ(3, 1)のMax Pooling層1つをつける。その後平滑化を行い、全結合層を2つ重ねてSoftmax関数により予測確率を出力する。出力層以外の活性化関数には全てReLUを、最適化関数にはAdadelta、損失関数にはCategorical Cross Entropyを用いた。

バッチサイズは32, エポック数は2048で学習を行った。

判断根拠については実装方法A, 実装方法Bと同様である。

## 2.5 実行環境

実装はGoogle Colaboratoryにて行った。言語はPython 3.10.12, 機械学習フレームワークにはTensorFlow 2.1 2.0を用いた。学習はGoogle Colaboratoryの提供するGPU(Tesla T4)によって実行した。

## 3. 結果と考察

### 3.1 実装方法A: 画像データを用いたCNNによる実装

筆者を正しく予測できた確率は92.3%であった。図2に学習経過の様子としてエポックごとの損失と精度を示す。図3には混同行列を示す。縦が正解値, 横が予測値である。

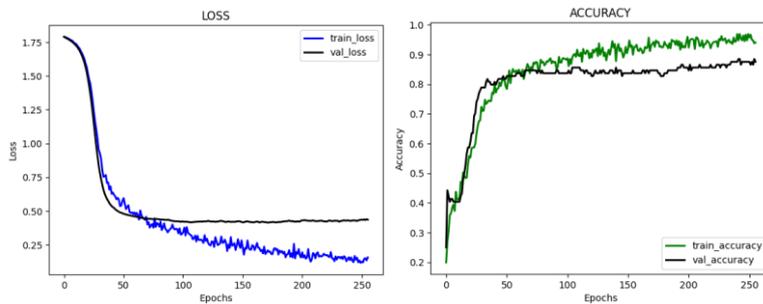


図2. 実装方法Aの学習経過

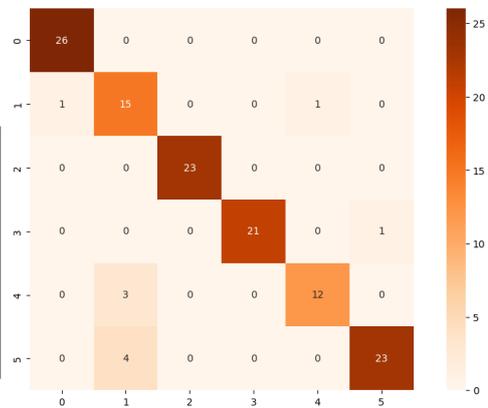


図3. 実装方法Aの混同行列

次に判断上の重要度を可視化した画像の一部を図4に示す。赤に近いほど重要度が高く, 紫に近いほど低い。

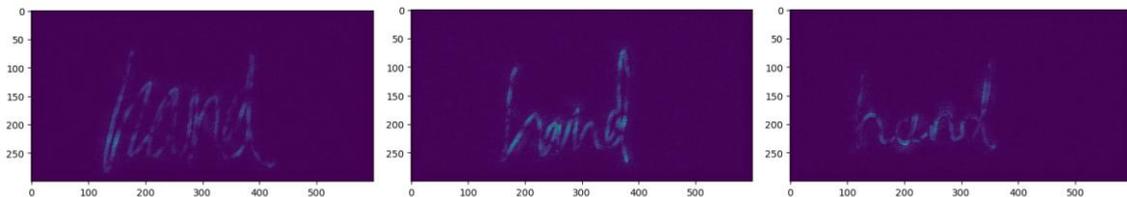


図4. 実装方法Aでの重要度を可視化した画像

文字の書かれた場所とそうでないところで差が出ているのは見て取れるが, 文字の中で実際にどこが重要かを解釈することは難しい。重要度の値に標準化や対数変換を施しても書いた部分の中に分かりやすい差は現れず, 重要度の比較はできなかった。

### 3.2 実装方法B : 時系列データを用いたLSTMによる実装

筆者の識別精度は96.9%であった。実装方法Aと同様に図5に学習経過の様子を, 図6に混同行列を示す。

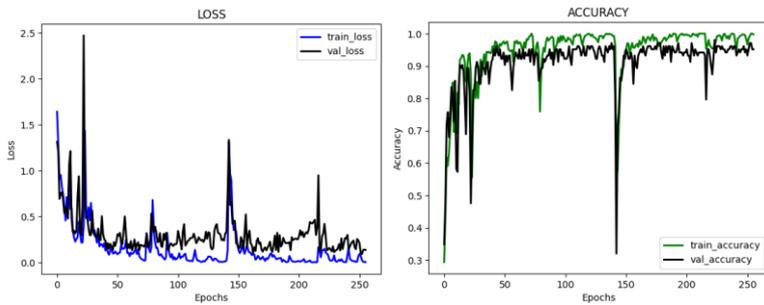


図5. 実装方法Bの学習経過

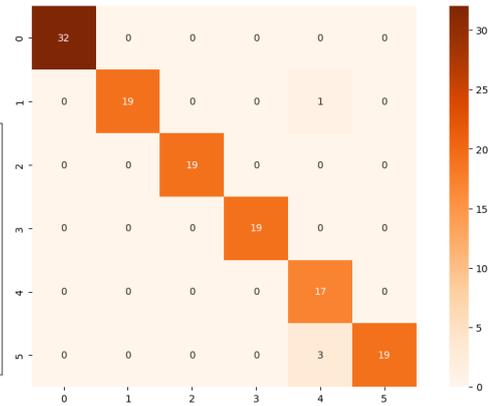


図6. 実装方法Bの混同行列

次に判断上の重要度を可視化した画像の一部を図7に示す。また、最後に欠けた「hand」とひらがな「ん」についても同様の学習と処理を行った結果を図8に示す。

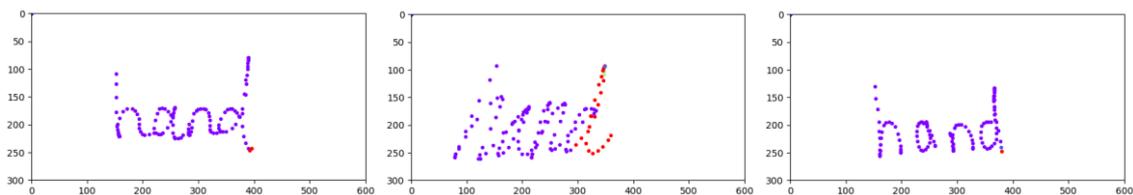


図7. 実装方法Bでの重要度を可視化した画像

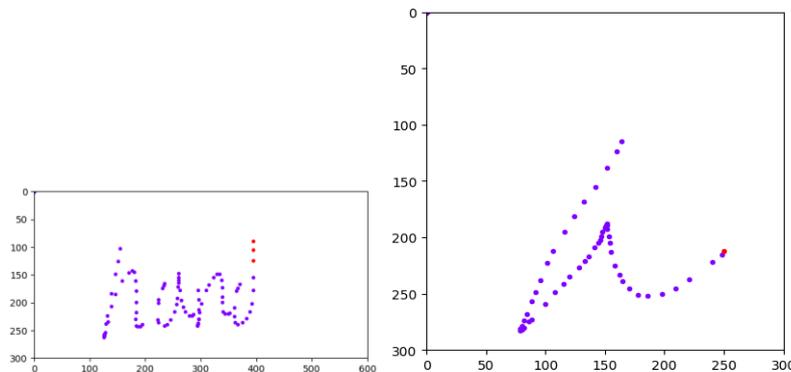


図8. 最後に欠けた「hand」と「ん」についての重要度を可視化した画像

図7を見ると、3つの例で全てにおいて「hand」の最後の部分の重要度が高くなっていることがわかる。実際、出力したほぼ全ての画像においてこの性質が見られた。この結果のみから判断すると、「hand」の最後の部分が判断上重要な部分であるか、あるいは今回とった手法自体に問題があるかという考察ができる。そこで、他の文字についても同じ実験を行った結果が図8である。どちらについても図7と共通の性質が見られるため、原因が手法にあることを特定できた。LSTMは短期記憶と長期記憶を独立にもつことによってRNNの勾配消失問題を緩和したものであるが、当実験における重要度の出力結果を見ると最後以外の勾配が消失していることがわかる。時系列内の前の部分と後ろの部分の勾配が対等に扱われていないという点で今回の手法には適していなかったことがわかる。

### 3.3 実装方法C : 時系列データを用いたCNNによる実装

識別精度は90.8%であった。実装方法Aと同様に図9に学習経過の様子を、図10に混同行列を示す。

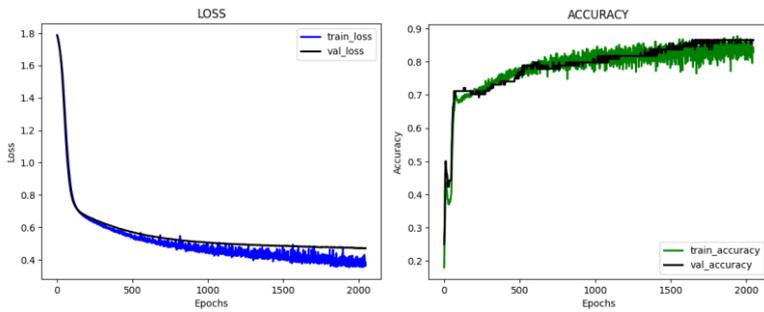


図9. 実装方法Cの学習経過

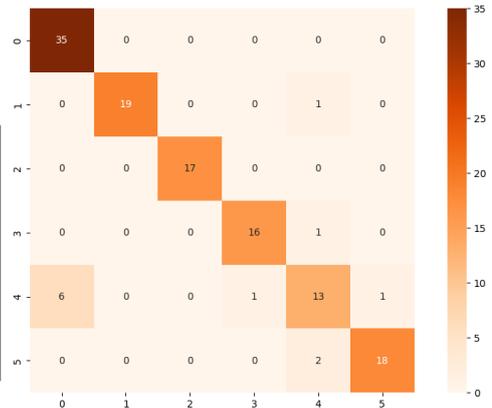


図10. 実装方法Cの混同行列

次に判断上の重要度を可視化した画像の一部を図11に示す。

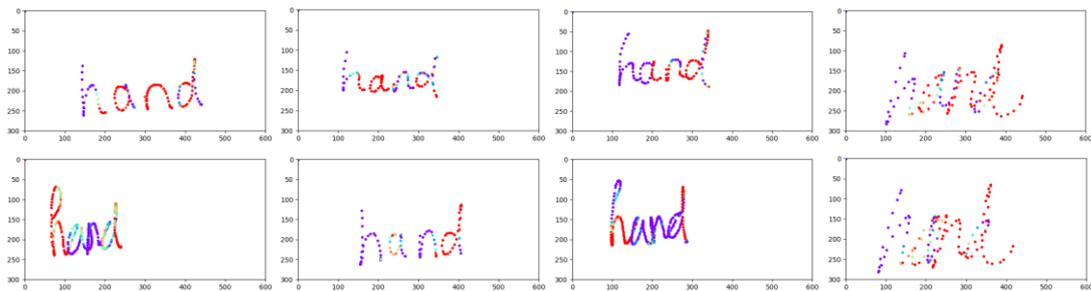


図11. 実装方法Cでの重要度を可視化した画像

時系列データを用いることで筆跡部分内での比較が可能となり、またCNNを用いることで勾配が消失することもなくなったため、判断根拠を可視化することができた。図11の各画像を見ると、線の伸びているところや丸まっているところなど、同一の文字の中でも場所によって色が分かれていることがわかる。しかし画像を見ただけではその全体的な傾向を掴むことはできず、結局筆跡鑑定AIの判断根拠は各画像に対して調べることはできても統一的に理解することが不可能な状況にある。

そこで、次の4. 追加実験 で示す実験を行った。

## 4. 追加実験

### 4.1 目的と仮説

実装方法Cにおける重要度を可視化したものからは全体的な判断根拠を掴むことができなかったため、重要度と人間にとって理解しやすい特徴量の関係を調べることで判断根拠に対する理解の深化を目指す。ここでは、「書く速さ」や「線の曲がり具合」と重要度の間に何か関係があるのではないかと仮説のもと、実験を行う。

### 4.2 方法

実装方法Cで作成したモデルを使用した。対象となる時系列データの各瞬間に関して、それに対応する書く速さや線の曲がり具合と重要度を散布図にそれぞれプロットし、これらの間にある関係を調べた。ここで時刻 $t$ における書く速さ $v_t$ と線の曲がり具合 $\tilde{\theta}_t$ は、時刻 $t$ におけるペン先の座標を $(x_t, y_t)$ とおいたとき次式で定義される。

$$v_t = \frac{1}{6} \sum_{k=t-5}^t \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}$$

$$\tilde{\theta}_t = \sum_{k=t-5}^t \frac{\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} \cdot \begin{pmatrix} x_k \\ y_k \end{pmatrix}}{\left| \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} \right| \left| \begin{pmatrix} x_k \\ y_k \end{pmatrix} \right|}$$

### 4.3 結果

得られた散布図の一例を図12, 13に示す。横軸はそれぞれ書く速さと線の曲がり具合, 縦軸はどちらも損失関数を微分した値である。

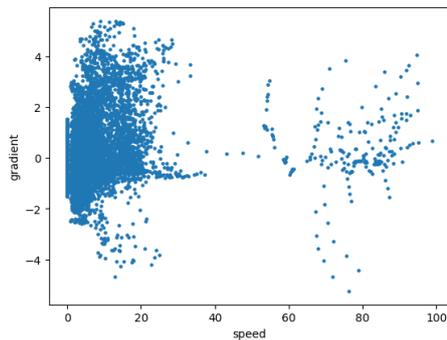


図12.書く速さと勾配の散布図

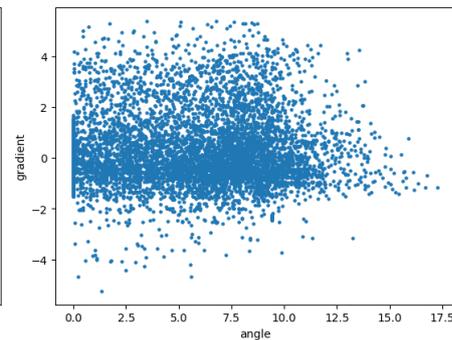


図13.線の曲がり具合と勾配の散布図

### 4.4 考察

どちらの散布図においても, それぞれの特徴量と勾配の間にある関係に明らかなものを見て取れない。単純な指標では判断の傾向を掴むことはできなかった。この先のアプローチとして, 抽出する特徴量の組み合わせやその重みを細かく調整して算出した指標と勾配の関係を逐一確認し, 判断の傾向をよく表すものを探すという方法が考えられるが, これは現実的ではない。判断根拠を人間の分かりやすい形で出力するシステムを開発するためにはさらなる研究が必要と考えられる。

## 5. まとめ

### 5.1 結論

本研究では, 画像データを用いたCNN, 時系列データを用いたLSTM, 時系列データを用いたCNNからなる3種の方法で筆跡鑑定AIを実装した。どの手法でも9割前後の精度を記録したが, 判断根拠の可視化に成功したのは時系列データとCNNを用いた実装方法Cのみである。またこのことについても, 画像単位の単純な可視化にこそ成功したものの, 全体的な判断の傾向をつかむことが困難という点で課題の残る結果となった。

### 5.2 今後の展望

今回実験に使用した筆跡は「hand」のみであったので, より汎用的なAIを作り, より一般的な実験結果を得るために対象とする文字の種類を広げる必要がある。判断根拠については, どのような形であれば人間が解釈しやすく, かつシステムとして実現可能性が高いかという点からさらに考察を進める必要がある。

### 謝辞

群馬大学数理データ科学教育研究センター講師の中村 賢治様には判断方法の可視化方法についての助言を, 指導教員の岡田 直之先生には論文執筆における助言や数多くの発表機会の提供をしていただきました。深く感謝致します。