

## 機械学習を用いたパターン認識による筆者識別

高橋 真奈茄<sup>1</sup> 小出 洋<sup>2</sup>

**概要:** コンピュータは高度な演算が可能である一方、人物の識別などは不得手とされている。このような識別における課題の一つとして、筆跡の筆者識別が挙げられる。本稿では、機械学習を用いたアプローチからコンピュータによる効果的な筆跡の筆者識別手法を提案し、視覚情報に基づく判断論理形成についての考察を行う。提案手法では、筆跡画像を幾何学的に解析し、階層型ニューラルネットワークを用いたパターン認識によって筆者を識別する。階層型ニューラルネットワークを用いることで、より柔軟な筆者識別を目指す。また、提案手法を実装し、実装したシステムによる筆者識別実験と、改良したシステムによる処理時間計測実験を実施した。筆者識別実験では、最良で78%の識別精度を得られた。処理時間計測実験では、処理速度が8.6倍に向上した。

**キーワード:** 筆跡, 機械学習, ニューラルネットワーク, バックプロパゲーション法, 画像解析

## Writer Identification by the Pattern Recognition with Machine Learning

**Abstract:** Although computers process a lot of tasks efficiently, they are weak in some problems like human recognition. One of these problems is a writer identification. In this manuscript, the authors propose an efficient pattern recognition method to identify a writer by using machine learning. The authors also give consideration to a logic to decide a author based on sight information. In The proposed method, makes an analyze of handwriting images geometrically first. And it identify a writer by using a pattern recognition with multi layer neural networks finally. Aim more flexible writer identification with using multi layer neural network. The authors implement the proposed methods on a multicore machine. And authors conduct experiments to evaluate the proposed method, and processing speed. The proposed method recognized a writer with 78 percent precision and the authors improved the processing speed of about 8.6 times.

**Keywords:** Handwriting, Machine Learning, Neural network, Back propagation, Image analysis

---

<sup>1</sup> 九州工業大学情報工学部知能情報工学科  
Department of Artificial Intelligence Kyushu Institute of Technology

<sup>2</sup> 九州工業大学大学院情報工学研究院情報創成工学研究

---

系  
Department of Creative Informatics Faculty of Computer Science and Systems Engineering Kyushu Institute of Technology

## 1. はじめに

本稿では、機械学習を用いたアプローチからコンピュータによる効果的な筆跡の筆者識別手法を提案し、提案手法の実装を行う。また、実装したシステムによる識別実験を行い、視覚情報に基づく判断論理形成について考察する。

近年、人工知能や汎用的な人型ロボットの研究開発が活発に行われており、対話を行うコンピュータの普及が始まっている。このような対話型コンピュータへの要求として、バイオメトリクスをはじめとする視覚情報に基づく識別技術が必要となることが予想される。しかし、このような分野において、コンピュータはその課題を効果的に解決できているとは言えない。そこで、課題の一つとして筆者識別に注目する。筆者識別は、筆者不明の筆跡が誰によって書かれたものであるか、その筆者を特定するものである。筆者識別は筆跡鑑定とも呼ばれ、専門家が犯罪捜査などに用いる。また、クレジットカードの利用伝票へのサインなどで個人認証の一種として用いられることもあり、我々の生活に身近な課題である。

筆者識別に注目した先行研究として、筆跡から個人性を抽出し、筆者識別を行ったものが挙げられる [1][2]。中村ら [1] は、書写技能に基づく特徴値を用いたオンライン筆者認識手法を提案した。中村らの研究では、字種を考慮しない特徴値を用いた場合に、筆者識別率は高いが偽の筆跡を誤って受理する誤照合率が高いことが分かっている。また、尺長ら [2] は、人間が比較的容易に、且つ、字種を考慮しない筆者識別を行える点に注目し、テクスチャ解析による新しい統計的手法を提案した。尺長らの研究により、テクスチャ解析の立場から高い筆者識別率を得られることが分かっている。これらの研究では、筆跡の特徴値を定義し、そこに現れる個人性を抽出することで、統計数理による手法を用いて筆者を識別している。一般的に、統計数理による手法では数理的に信頼できるだけの学習データが必要である。

本稿では、字間などの特徴値やテクスチャ解析

が筆者識別に有効であることから、幾何学的な画像解析によって機械的な特徴値の抽出を行う。しかし、誤照合率の高さや、柔軟性の低さを補うために、統計数理による手法ではなく、人工知能による手法を用いることを提案する。筆者識別のようなパターン認識問題には、一般的に、人工知能の一分野である機械学習が用いられる。特に、階層型ニューラルネットワークは、既に顔画像認識に有効であることが分かっている [3] 他、様々な課題に用いられている。そこで、筆者識別に階層型ニューラルネットワークを用いる。

## 2. 筆跡画像の解析

### 2.1 特徴値の抽出

筆跡からどのように特徴値を抽出するかについては、既にいくつかの研究がなされている [1][2]。中村ら [1] の研究により、字の大きさや配置に関する特徴値のみを用いた場合でも、個人性を得られることが分かっている。また、尺長ら [2] の研究により、字種を考慮しないテクスチャ解析による特徴値の抽出から、個人性を得られることが分かっている。そこで、階層型ニューラルネットワークを用いることで柔軟な識別ができると予想し、字種情報を考慮しない単純な特徴値を抽出し、それを用いる。単純な特徴値を用いた筆者識別が可能であれば、複雑な特徴値からより高い精度での識別ができると予想できるためである。

一般的に、画像解析における画像特徴は以下の4つに分類できる [4]。

- (a) 濃度, 色彩, マルチスペクトル情報
- (b) テクスチャ情報
- (c) 位置, 形状情報
- (d) 空間的構造・関係情報

この分類から、筆跡画像の特徴値として濃度情報と位置・形状情報を用いる。人間は筆跡の濃度によってある程度の識別が行えることから濃度情報を、字の大きさや配置は個人性を表すことから位置、形状情報を用いる。また、テクスチャ情報は個人性を表す有効な情報であると述べたが、尺長ら [2] は、既存の手法によって筆跡画像をテクスチャ解析するには不具合が生じるために新しい解

析手法を提案している。本稿では、まず、単純な特徴値からの識別について検討するため、これを用いない。空間的構造・関係情報も同様に、多様な字種を一律に扱えないと予想されるため、用いない。具体的には、以下の3つを特徴値として抽出する。

- (1) 濃度ヒストグラム
- (2) 画素値
- (3) 文字の占有率

次に、これらの特徴値について述べる。

### 2.1.1 濃度ヒストグラム

画像解析におけるヒストグラムは各濃度値に対して画像中でその濃度値をもつ画素の数の程度を表すグラフであり [4]、横軸に濃度値、縦軸に度数をとる。ヒストグラムは 256 段階の濃度値について計算されるため、256 個の値から構成される。ここで、 $p = 256$  とおく。ヒストグラムは一般的に画像全体に対して計算されるものである。しかし、筆跡画像は高い画素値が大部分であるため、画像全体のヒストグラムを計算することは特徴の精度の点で問題がある。そこで、二値画像をマスクにしてヒストグラムを計算した。これにより、筆跡の字画部分についてのみヒストグラムを計測できる。

### 2.1.2 画素値

筆跡画像は黒い字画と白い記入領域から構成される線画像であるため、位置・形状情報の特徴として二値画像の画素値そのものを用いた。数値の単純化のために画素値 0 を 1、画素値 255 を 0 とした。単純化した画素値をそのまま用いるとその要素数は膨大であるため、データ圧縮のために二値化した画素値で画像の 1 行を 1 つの二進数とみなして十進数へ基数変換を行った。筆跡画像の大きさは  $559 \times 163$  ピクセルであるから、十進変換した画素値は 163 個の数値で構成される。ここで、 $q = 163$  とおく。

### 2.1.3 文字の占有率

筆跡の太さ・大きさを表す特徴として、筆跡画像のうち字画を表す画素の割合を計算した。これを文字の占有率と定義する。文字の占有率は、筆跡画像一つにつき 1 個の数値から構成される。ここで、 $r = 1$  とおく。

## 3. 機械学習を用いたパターン認識

### 3.1 ニューラルネットワーク

本稿では、機械学習の手法の一つであるニューラルネットワークを用いて筆者識別を行った。ニューラルネットワークは、モデル化したニューロン（以下、ニューロンと呼ぶ）を多数つなげてネットワークを構成し、その中で信号伝達を繰り返すことによって情報処理を行うアルゴリズムである [5]。ニューラルネットワークは学習機能をもつため、判断論理を自ら形成する。図 1 にニューロンを示す。ニューロンは入力の重み付き総和が適当な閾値で発火して出力される。発火の関数は様々あるが、シグモイド関数を用いた。式 (1) にシグモイド関数を示す。シグモイド関数は出力が滑らかに変化し微分しても連続な関数であり、数学的に扱いやすいためニューラルネットワークでよく用いられている。

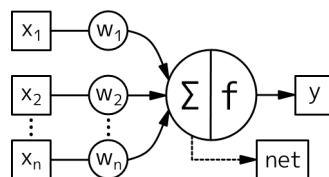


図 1 モデル化したニューロン  
Fig.1: A model for neuron.

$$f(x) = \frac{1}{1 + \exp^{-x}} \quad (1)$$

次に、ニューロンによるネットワークの構成について述べる。ネットワークの構成には階層型と相互結合型があるが、パターン認識に向いているとされる階層型を用いた。階層型ネットワークはその名の通り層状のネットワークで、入力層、中間層、出力層から構成される。入力層は入力信号に処理を加えずに中間層へ伝達する層である。中間層は前後の層に対して結合をもち、信号伝達をすることで情報処理を行う層である。出力層はニューラルネットワークの出力結果を得るための最終層である。階層型ネットワークは隣り合う層のニューロン間だけに結合をもち、演算も一定方向

で行われる。入力層から出力層へ向かう演算は順方向演算、出力層から入力層へ向かう演算は逆方向演算と呼ばれる。図2に階層型ニューラルネットワークを示す。ここで、 $x_i$  は入力層  $i$  番目の入力信号の要素、 $m_j$  は中間層  $j$  番目のニューロンの出力、 $o_k$  は出力層  $k$  番目のニューロンの出力、 $v_{i,j}$  は入力層  $i$  番目ニューロンから中間層  $j$  番目のニューロンとの間の結合重みを、 $w_{j,k}$  は中間層  $j$  番目のニューロンから出力層  $k$  番目のニューロンとの間の結合重みを表す。

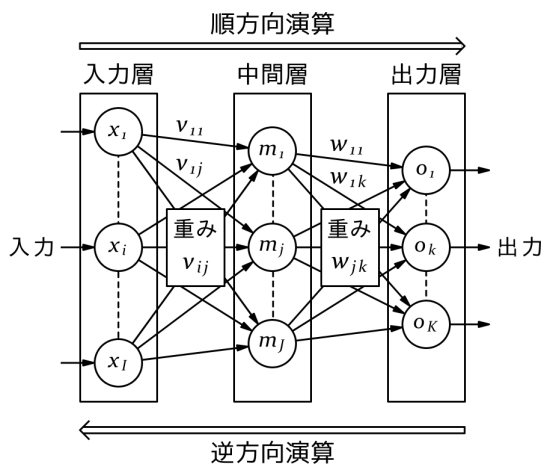


図2 階層型ニューラルネットワーク  
Fig.2: A multilayered neural network.

### 3.2 ニューラルネットワークの学習

ニューラルネットワークの学習手法には様々あるが、パターン認識のような問題には教師あり学習によるニューロン間の結合重みを調整する手法が適している [5]。教師あり学習は既知の入力信号とそれに対する正しい出力信号（教師信号と呼ぶ）の組を用いて行う学習である。本稿では、教師あり学習の代表的アルゴリズムであるバックプロパゲーション法 [5] を用いた。

バックプロパゲーション法は順方向演算と逆方向演算の繰り返しによってニューロン間の結合の重みを調整するアルゴリズムである。図3にバックプロパゲーション法の学習モデルを示す。ここで、 $t_k$  は出力層  $k$  番目に対する教師信号の要素を表す。ここで、順方向演算とそれに関する処理を

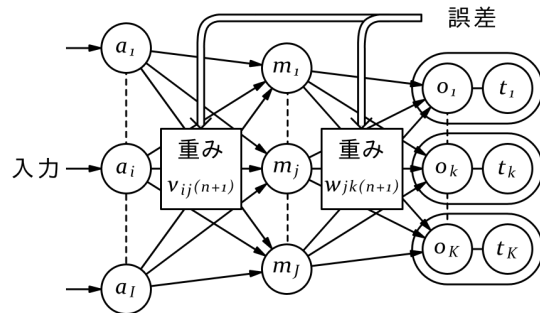


図3 バックプロパゲーション法の学習モデル  
Fig.3: A learning model of the back propagation.

行う過程を順方向過程、逆方向演算とそれに関する処理を行う過程を逆方向過程と呼ぶ。順方向過程では、入力信号を与えて順方向演算によりニューラルネットワークの出力を得る。学習時には入力信号と組になる教師信号が与えられているため、ここでの出力と教師信号との間の誤差を算出する。

逆方向過程では、順方向過程で得られた誤差を用いてニューロン間の結合重みを調整する。層間の新しい結合重みは誤差と重みの修正量を使って求める。誤差は、ここから学習信号を求めて伝搬する。出力層  $k$  番目のニューロンの学習信号は次の式から求める。ここで、 $f(x)$  はシグモイド関数、 $\sigma_k$  は出力層  $k$  番目のニューロンの重み付き総和を表す。

$$\delta_k = (t_k - o_k) f'(\sigma_k) \quad (2)$$

新しい結合重みは学習信号と重みの修正量を用いて次の式から求める。ここで、 $\Delta w_{j,k}(t)$  は現在の重みの修正量、 $\eta$  は学習係数、 $\alpha$  は収束係数、 $\delta_k$  は後の層の  $k$  番目のニューロンからの学習信号を表す。

$$\Delta w_{j,k}(t) = \eta \delta_k \sigma_j + \alpha \Delta w_{j,k}(t-1) \quad (3)$$

$$w_{j,k}(t+1) = w_{j,k}(t) + \Delta w_{j,k}(t) \quad (4)$$

中間層以降の学習信号は式 (2) で誤差  $t_k - o_k$  を学習信号  $\delta_k$  の重み付き総和と置き換えて求める。新しい結合重みはこのように求めた学習信号で式 (3) 式 (4) を用いて求める。

### 4. 提案手法の実装

これまで述べたような手法を実装し、筆者識別

システムを開発した。本システムは、入力された筆跡の筆者を、既知の筆者の中から特定するものである。筆者識別を行うためには既知の筆跡を用いた学習を必要とする。学習済みのシステムに未知の筆跡を入力すると、筆者識別を行う。既知の筆者は3名として学習・識別を行う。システム全体のフローチャートを図4に示す。

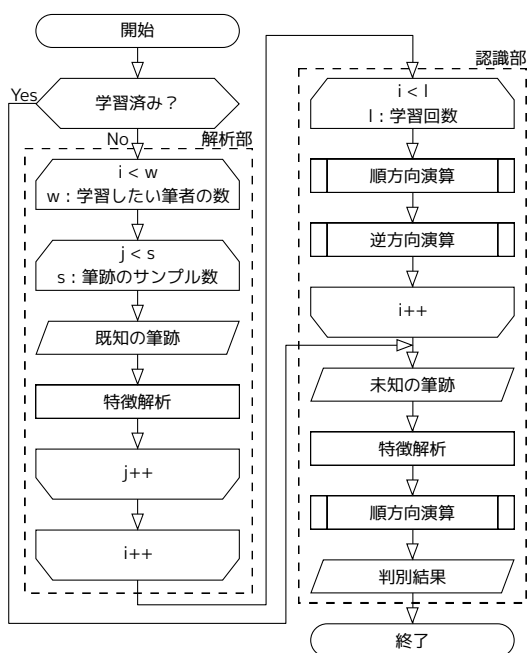


図4 筆者識別システムのフローチャート  
Fig.4: A flowchart of the writer identification system.

筆跡は独自に収集した筆跡データを用いる。図5に筆跡データの一例を示す。

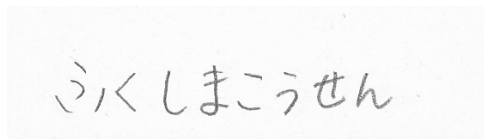


図5 筆跡画像  
Fig.5: A handwriting image.

#### 4.1 解析部

解析部では筆跡画像を解析し、特徴値を抽出をする。抽出する特徴値は(1)濃度ヒストグラム  $p=256$  個、(2)画素値  $q=163$  個、(3)文字の占有率  $r=1$  個の3種で、計  $p+q+r=420$  個である。次に、特徴抽出の具体的な処理について述べる。

まず、筆跡画像に対して、グレースケール化、平滑化を行う。グレースケール化は、RGB色彩画像を濃度値のみをもつグレースケール画像に変換する処理である。平滑化は、画像における雑音が高周波成分を多く含むことに着目し、空間周波数の協調処理によって画像の雑音を除去する処理である[4]。平滑化の手法には様々あるが、ガウシアンフィルタを用いた。2つの処理には、OpenCVの関数[6]を用いた。2つの処理を行った状態を、状態Aとする。状態Aの筆跡画像に対して、二値化を行う。二値化は、閾値に従ってグレースケール画像を白と黒の画素のみで構成される画像に変換する処理である。二値化には、OpenCVの関数[6]を用いた。状態Aの筆跡画像に二値化を行った状態を、状態Bとする。ここで、状態Aと状態Bの筆跡画像を用いて、濃度ヒストグラムと文字の占有率の抽出を行う。濃度ヒストグラムは、状態Bの筆跡画像をマスクとして状態Aの筆跡画像から抽出する。文字の占有率は、状態Bの筆跡画像から抽出する。最後に、状態Bの筆跡画像に対して位置調整を行う。位置調整は、筆記位置の違いによる筆跡の位置・形状情報のノイズを取り除くために筆跡の中心を画像の中心へ平行移動させる操作である。筆跡の中心は二値画像中で黒の画素が現れる最も小さなx座標  $s_x$  と y座標  $s_y$  の組、最も大きなx座標  $e_x$  と y座標  $e_y$  の組を見つけ出すことで算出した。位置調整には、独自に実装を行った関数を用いた。状態Bの筆跡画像に二値化を行った状態を、状態Cとする。画素値は、状態Cの筆跡画像から抽出する。

以上の処理によって求められた特徴値を、次の認識部への入力信号とする。

#### 4.2 認識部

認識部では解析部で求めた特徴値を用いて、

ニューラルネットワークによる筆者識別を行う。認識部は、学習機能と識別機能を持つ。前節で述べたように入力信号は 420 個の数値で構成されるため、入力層ニューロン数は 420 個である。出力層ニューロンは筆者と一対一対応をとる。既知の筆者は 3 名であるから、出力層ニューロン数は 3 個である。識別結果は、入力された筆跡と既知の筆者との一致度として出力される。一致度は 0 から 1 までの浮動小数で、数値が高いほど一致度も高い。識別の成功は「入力した筆跡の筆者と一致度が最も高いニューロンが一致して、かつ、その値が他のニューロンの一致度より十分に高い」と定義する。ここで、出力の最大値  $max1$  と二番目に大きな値  $max2$  が次の条件を満たすとき、最大値が他の値より十分に大きいと判断する。

$$\frac{max1 - max2}{max1} > 0.6 \quad (5)$$

図 6 に学習済みニューラルネットワークによる筆者識別のモデル図を示す。

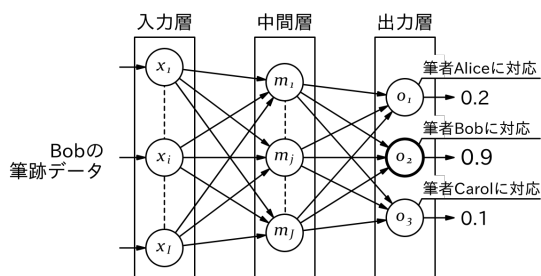


図 6 学習済みのニューラルネットワークによる筆者識別のモデル図

Fig.6: A model diagram of writer identification by the learnt neural network.

## 5. ニューラルネットワークの高速化

ニューラルネットワークは多数のニューロンから構成されるネットワーク内で信号伝達を繰り返すことで情報処理を行うアルゴリズムである。そのため、膨大な数値データによる多数の繰り返し処理を行う。また、ニューロン単体の動きは単純で独立しているため、並列処理に向いている。このような特性上、高速化や並列化を考慮しないプログラムでは処理速度が大幅に遅くなる。これはシス

テムの実用性にも支障があるほか、実験を効率的に行えない問題がある。そこで、ニューラルネットワークの特性に着目して処理の高速化を図る。

### 5.1 プログラムの書き換え

高速化前のプログラムは C# で記述されており、処理の高速化を図る工夫はなされていなかった。また、Windows 上で動作する GUI システムであり、研究を進めるためには不都合である。そこで、C++ によってプログラムの書き換えと実行環境の移行を行った。C++ は一般的に高速であると言われており、C++ バージョン 11 以降は標準ライブラリで並列処理をサポートしている。また、C++ コンパイラは様々な環境上で動作するため、効率的な処理が可能なマシン上で実験を行うことができる。

これらの理由から、C++11 に準拠したプログラムによりシステムの実装を行った。コンパイラは clang/clang++ バージョン 7.0.0 を使い、C++11 の機能を有効にするため `-std=gnu++11` オプションをつけてコンパイルを行った。

### 5.2 高速化の実装

#### 5.2.1 メモリアクセス

ニューラルネットワークの実装は配列へのアクセスを多用する。このため、キャッシュ効率を考慮し、メモリへのアクセス時間短縮するような実装を行うことで処理時間を短縮できることが考えられる。図 7 にループ入れ替えによる配列へのアクセス効率化の実装例を示す。順方向演算中にループ入れ替えを適用することで、配列アクセスを効率化した。

```

- for (int i=0; i<lim; i++) {
-     for (int j=0; j<lim; j++) {
+ for (int j=0; j<lim; j++) {
+     for (int i=0; i<lim; i++) {
        sum += vec.at(j).at(i);
    }
}

```

図 7 ループ入れ替えによる配列へのアクセス効率化  
Fig.7: Better efficient access to an array by loop exchange.

### 5.2.2 演算の並列化

ニューラルネットワークは、多数のニューロンについて重み付き総和などの単純な演算を行う。各ニューロンは独立して演算を行うため、データの競合が発生しづらい。これは並列処理を行うのに都合が良く、順方向演算や逆方向演算の並列化によって処理速度が向上すると予想される。そこで、スレッド並列による順方向演算の並列化を行った。並列化は2つのスレッドにより行った。スレッドに中間層ニューロンの半分ずつを割り当て、各スレッドは割り当てられたニューロンについて重み付き総和と出力を計算する。

### 5.2.3 ファイル入出力

筆者識別システムでは画像ファイルを読み込んで解析を行う。ファイル入出力は数値の演算などに比べて非常に遅い処理であるから、その回数は最低限に抑えたい。そこで、ファイル読み込みが最低限の回数となるように改良した。高速化前のプログラムでは、学習サイクル中で筆跡画像の解析を行うクラスのインスタンスを生成していた。画像解析クラスのコンストラクタではファイルを読み込んでいるから、インスタンスが生成されるたびにファイルの入出力を行うことになる。そこで、学習サイクルの外で先にインスタンスを生成して初期化を行っておき、学習サイクル中では入力信号の計算のみを行う。図8にファイル入出力回数削減の実装を示す。

```
+ for (int j=0; j<numUnit0; j++) {
+   img.at(j).resize(inputpath.at(j).size());
+   for (int k=0; k<inputpath.at(j).size(); k++) {
+     img.at(j).at(k) = new ImageAnalyzer(inputpath.at(j).at(k));
+     img.at(j).at(k)->init();
+   }
+ }

for (int i=0; i<cycle; i++) {
  for (int j=0; j<numUnit0; j++) {
    for (int k=0; k<inputpath.at(j).size(); k++) {
-     ImageAnalyzer img(inputpath.at(j).at(k));
-     img.init();
+     signalInput = img.formatInputSignal();
+     signalInput = img.at(j).at(k)->formatInputSignal();
      script.learn(signalTeach, signalInput);
    }
  }
}
```

図8 ファイル入出力回数の削減

Fig.8: Reduce of the number of access to file I/O.

## 6. 筆者識別実験

提案手法によって筆者の識別が可能であるかを検証するため、筆者識別システムを用いた筆者識別実験を行った。

### 6.1 実験方法

実験には、独自に収集した筆跡データを用いた。筆跡データは、規定の用紙の枠内に0.5mmの黒のシャープペンシルを用いて、「できるだけ普段通りに」6個筆記したものを29名分収集した。さらに、条件を「できるだけ丁寧に」「できるだけ普段通りに」「少し崩して」に増やし、それぞれ6個ずつ筆記したものを6名分収集した。収集した筆跡データはスキャナで読み取り、枠内を559×163ピクセルずつ切り取る。スキャナでの読み取りは、A4サイズ、300dpi、JPEG形式で行う。

実験は、「できるだけ普段通りに」筆記した筆跡データのみを既知の筆跡として学習に用いる実験Aと、3種の条件で筆記した筆跡データを既知の筆跡として学習に用いる実験Bの2つを行った。実験Aでは、筆者1名につき3個の筆跡を既知の筆跡として学習に使用し、残り3個の筆跡を未知の筆跡として識別試験に使用した。実験Bでは、筆者1名につき各条件で3個ずつの筆跡を既知の筆跡として学習に使用し、残り9個の筆跡を未知の筆跡として識別試験に使用した。なお、実験は収集した筆跡データのうち3名分を用いて行った。

ニューラルネットワークは、中間層ニューロン数や学習回数を変化させることで情報処理能力が変化することが一般的に知られている。そこで、ニューラルネットワークのパラメータの変化による識別結果の変化を検証するため、中間層数200、520と変化させる。また、学習回数は1000回、5000回、10000回と変化させる。なお、実験Bでは中間層数200、学習回数1000回、5000回のみ実験を行った。

### 6.2 実験結果

筆者識別実験の実験結果を表1、2にそれぞれ

示す。

表 1 実験 A での識別率 [%]

Table1: Identification rate on the experiment A.

		中間層ニューロン数	
		200	520
学習回数	1000	61	61
	5000	67	61
	10000	67	78

表 2 実験 B での識別率 [%]

Table2: Identification rate on the experiment B.

		中間層ニューロン数
		200
学習回数	1000	69
	5000	61

### 6.3 検討

筆者識別実験を行った結果、最良で 78% の識別率が得られた。これは、先行研究と比べると大幅に低い識別率である。しかし、実験中に、筆跡毎の識別率で有意な差が認識できた。実験 A における筆跡毎の識別率を表 3 に示す。ここでは、3 つの

表 3 実験 A における筆跡毎の識別率 [%]

Table3: Identification rate for each handwriting on the experiment A.

筆者 K	筆者 T	筆者 Y
17	42	100
100	100	67
33	50	83

筆跡でそれぞれ 100% の精度が得られている。これは、ニューラルネットワークが筆跡の特徴値に現れる個人性を捉えている可能性を示唆している。また、表 1 より、学習回数の増加にしたがって識別率も向上していることが読み取れる。加えて、表 3 に中程度の精度で識別できる筆跡があることから、学習を重ねることでより適切に個人性を抽出できていると考えられる。既知の筆跡の特徴値からニューラルネットワークが個人性を学習できて

いる可能性があるということは、提案手法のアプローチは有効であると言える。

また、実験 B では、意図的に同一筆者の筆跡に違いをもたせることで筆跡に現れる個人性をより効果的に学習できると予想したが、意図的な違いを含まない実験 B と識別率に有意な差は確認できなかった。しかし、中間層ニューロン数と学習回数を増やした条件での識別実験は、学習サンプルの増加による処理時間増大のために行っていない。未確認の条件には実験 A で最良の結果が得られたものが含まれるため、条件を追加して筆者識別実験を行うことで、より良い結果が得られることが予想される。

## 7. 処理時間計測実験

素直な実装に比べて処理速度がどれだけ向上したかを検証するため、高速化前後の筆者識別システムを用いた処理時間計測実験を行った。

### 7.1 実験方法

筆者識別実験の実験 B と同様の手順で実験を行い、既知の筆跡の学習に要する処理時間の計測を行った。中間層ニューロン数を 200、学習回数を 1000 に設定し、処理時間をそれぞれ 3 回計測した。実験結果は、3 回の計測の平均とした。表 4、5 にプログラムの実行環境を示す。なお、高速化前のシステムの実行環境は仮想マシン上に構築した。

表 4 高速化前のシステムの実行環境

Table4: An execution environment of the original system.

プロセッサ	Intel Corei5 2.6GHz
メモリ	4GB
OS	Windows 8.1
実装	C#

### 7.2 実験結果

処理時間計測実験の実験結果を、図 9 に示す。プログラムは次の順で改良し、それぞれの時点で処理速度を計測した。

(1) C++への書き換え



表 5 高速化後のシステムの実行環境

Table5: An execution environment of the speedy system.

プロセッサ	Intel Corei5 2.6GHz
メモリ	16GB
OS	OS X 10.10.5
実装	C++ 11

- (2) ループ入れ替え
- (3) コンパイラの最適化オプション適用
- (4) 並列化
- (5) ファイル入出力の回数を削減

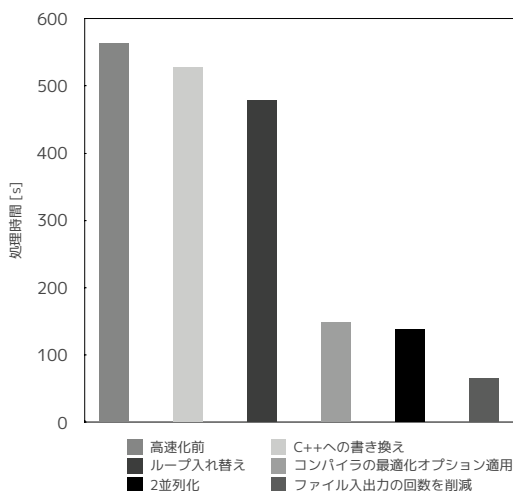


図 9 処理速度の計測結果

Fig.9: The elapse time of the proposed method.

図 9 より、最終的に、高速化前と比べて処理速度が 8.6 倍に向上した。また、(1) から (5) の高速化手法を適用することで、処理速度はそれぞれ (1)1.1 倍、(2)1.1 倍、(3)3.2 倍、(4)1.1 倍、(5)2.1 倍に向上した。

### 7.3 検討

実験結果より、適用前後で最も効果的であったのは、コンパイラの最適化オプションの適用であった。最適化オプションは、コンパイラが提供しているほとんどの最適化を行うものを選択した。コンパイラの最適化は演算命令の省略など行うものであるから、ほとんどがループ処理と演算で構成

されるニューラルネットワークには効果的であったと考えられる。次いで、ファイル入出力回数の削減が効果的であった。この結果は、ファイルの入出力は演算と比べて遅い処理であるからその回数を削減することで高速化が行える、という予想と合致する。ループの入れ替え、並列化はどちらも処理速度の向上は 1.1 倍にとどまった。まず、ループの入れ替えについて検討する。実際の実装では、ループ入れ替えに加えて処理手順を変えないために if 文を追加している。条件分岐の処理時間が追加されるため、ループ入れ替えの効果と相殺され、高速化の効果が低くなったと考えられる。次に、並列化について検討する。2 スレッドによる並列化を行ったため、理想的には順方向演算が 2 倍に高速化される。並列化の後に行ったファイル入出力に関する高速化などが効果的であったことから、処理時間に占める順方向演算の割合が小さかったことが素因の一つであると考えられる。また、スレッドの生成やスレッドへのタスク割り当てによって処理が増加するため、処理時間は増加する。順方向演算の中でも中間層ニューロン出力の演算と出力層ニューロン出力の演算を別々に並列化したため、理想的な並列化が行えなかったことが考えられる。

## 8. 考察

ニューラルネットワークを用いて単純な画像特徴から筆跡に現れる個性を抽出する手法が有効であるならば、単純な画像特徴からのさまざまな識別が可能であると予想される。本稿では筆跡に現れる個性を識別したが、視覚情報から単純な特徴を解析し、それに基づく人間の感覚を学習させることは様々な課題に適用できる。また、同じデータについて学習を繰り返すことで正しい識別結果に近づくことから、ニューラルネットワークは自身の判断論理を修正しながら、柔軟な判断論理の形成ができていると言える。

## 9. おわりに

幾何学的な画像解析による特徴値抽出と階層型ニューラルネットワークにより、既知の筆跡を学

習して未知の筆跡に対して筆者識別を行うシステムの開発を行った。このシステムによる筆者識別実験では、最良で78%の識別率が得られた。また、高速化したシステムによる処理時間計測実験では、高速化前後で8.6倍に処理速度が向上した。筆者の識別率は決して高い精度ではないが、筆跡毎の識別率は最良で100%であり、提案手法のアプローチが有効である可能性を示した。

研究には大きな余地が残されていると言えるため、今後の展望として次の4つを挙げる。(1) 学習サンプルを50個程度に増やすこと、(2) 照合実験を行うこと、(3) 特徴値の再検討を行うこと、(4) 処理の並列化を行うこと。

対話型コンピュータへの要求としての視覚情報に基づく識別問題は多岐にわたり、ニューラルネットワークは単純な画像特徴でそれらに汎用的に対応できる。ニューラルネットワークを用いて視覚情報に基づく識別課題の一つについて研究することで、同様の手法を用いた他の課題解決への考察を行う。

#### 参考文献

- [1] 中村, 豊田: 書写技能に基づく筆跡に現れる個性の抽出, 電子情報通信学会論文誌 J77-D-2 巻3号, pp510-518 (1994)
- [2] 尺長, 金子, 淀川: 2次統計量の線分スペクトル分解—テキストチャ生からの手書き文字個人性情報の抽出—, 信学論 (D), J67-D-4, pp488-495 (1984)
- [3] 吉田 昌弘: 階層型ニューラルネットワークのローカル学習アルゴリズムと顔画像認識への応用, 静岡大学博士論文 (2005)
- [4] 田村 秀行: コンピュータ画像処理, オーム社 (2004)
- [5] 萩原 将文: ニューロ・ファジィ・遺伝的アルゴリズム, 産業図書 (2011)
- [6] imgproc. 画像処理 - opencv 2.2 documentation: [http://opencv.jp/opencv-2svn/cpp/imgproc.\\_image\\_processing.html](http://opencv.jp/opencv-2svn/cpp/imgproc._image_processing.html), (2015-11-10)
- [7] 高橋, 大槻, 小泉: パターン認識による筆跡の筆者判別, 福島高専卒業論文 (2013)