

平成12年度  
学士學位論文

**ニューラルネットワークを用いた掌紋認識の有用性に関する研究**

**Research on utility of palm print recognition  
by neural net works**

所属：情報システム工学科  
学籍番号：1010410  
氏名：千屋貴裕  
指導教員：竹田史章教授  
提出日：2001.2.5

## 要旨

これまでの個人認証には指紋を用いることが多かったが、最近では犯罪者の中で指紋をつぶす者があらわれるなど多種多様な問題が提起されている。本研究では、より安定し、偽造されにくい掌紋を取り上げて個人認証システムに着目する。とくに、掌紋画像の学習、識別にはパターン識別に高度な認証能力を発揮するニューラルネットワークを用い、その有用性をコンピュータシミュレーションで示す。

## Abstract

The fingerprint is often used for a current individual attestation. However, the criminal has the thing to erase my fingerprint. In this research, it is stable, and pays attention to the individual attestation system which uses palmprint that the counterfeited thing is difficult. The neural net work by which a highly developed attestation ability is demonstrated is used for the pattern recognition for the experiment. Then, utility is shown using the computer simulation.

1章	はじめに.....	1
2章	個人認証とは.....	4
2.1	バイオメトリクスを用いた個人認証.....	4
2.2	掌紋による個人認証.....	4
3章	ニューロ識別システム.....	5
3.1	システムの主な機能.....	5
3.2	ニューラルネットワーク.....	5
3.2.1	ニューラルネットワーク.....	6
3.2.2	ネットワークの構成.....	6
3.2.3	情報の伝達.....	7
3.2.4	中間層・出力層の入出力関数.....	8
3.2.5	学習アルゴリズム.....	9
3.2.6	学習パラメータ.....	11
3.3	学習システムの環境.....	12
3.3.1	データベース設定ファイル.....	12
3.3.2	環境設定ファイル.....	12
3.4	学習システムの動作.....	14
3.4.1	学習データ抽出 / 追加.....	14
3.4.2	マスクファイル作成.....	14
3.4.3	スラブファイル作成.....	17
3.4.4	学習ファイル作成.....	19
3.4.5	教師ファイル作成.....	19
4章	ニューロ識別システムを用いた実験.....	22
4.1	実験目的.....	22
4.2	実験機器.....	22
4.3	実験手順.....	23
4.3.1	実験の流れ.....	23
4.3.2	画像採取.....	23
4.3.3	学習.....	26
4.3.4	評価.....	29
4.4	実験条件.....	30
4.5	実験結果.....	31
5章	まとめ.....	33
6章	謝辞.....	34
7章	参考文献.....	35

# 1章 はじめに

昨今のネットワーク、マルチメディアなどの技術のめざましい発達に伴い、郵政省のWG6プロジェクトに代表されるように個人認証と言うものに注目が集まっている。とくに盗むことのできない完全な個人認証の必要性が高まってきている。個人認証の活用場所について代表的なものとしては、銀行における個人認証、クレジットカードにおける個人認証、アクセス管理における個人認証、入退場システムにおける個人認証などがあげられる<sup>(1)</sup>。

従来、セキュリティを実現するシステムの多くは、ユーザーを特定するためのユーザー名、すなわちIDと、それを入力したのが本人であることを確認するためのパスワードの組み合わせを用いてきた。また、銀行のカードなどのように、ID入力かわりにカードを用いる場合もあり、パスワードが数字の場合は暗証番号と呼ばれたりする。これらのシステムでは、パスワードは本人しか知らない情報であるということ根拠にしてユーザー認証を行うが、パスワードを盗み見られて他人になりすまされる危険がある上に、忘れてしまえば情報にアクセスできなくなる不便さがある。また、最近、様々なシステムでパスワードを要求されることがおおいが、同一のパスワードを使うのは危険であるため複数の異なったパスワードを記憶しなければならず、ユーザーの負担は大きなものとなってきている。さらに、IDカードを併用する方式は、セキュリティが向上する反面、カードを忘れたり紛失してしまうと情報アクセスができなくなる不便さがあるだけでなく、パスワードを解読されて不正使用される事件も発生している。

そこで、これらの問題を解決するため、本人であることをバイオメトリクス(生物学的な特徴)を用いて判別しようという方式が昨今注目を集めている。バイオメトリクスは全ての人で異なり、かつ年月がたっても変化しないような個人特徴でなければならない。バイオメトリクスは、大きく2つに分類され、本人の意思で変えることができない特徴点に注目するものと、恣意的に変えることができる特徴に注目するものがある。

前者には、顔貌、網膜、虹彩、耳、指紋、掌紋、掌形、指形、手の甲の血管模様などが該当する。後者には、声紋、署名(筆跡)などが該当する。ここで研究・実用化が進んでいるバイオメトリクスの例をいくつか紹介する。

- 虹彩(アイリス) - 黒目のうち瞳孔を囲む放射状の筋肉の表面にある模様を利用する。外部から見えやすく非接触で撮像できるという長所を持っている。精度は高い反面、カメラ系などのシステムが高価なため入門管理用とが主であったが、今後小型化・低価格化による情報セキュリティ分野への展開も図られて

いる。

- 網膜血管 - 眼球内部にある網膜上の血管パターンを用いる。認証精度は高いとされるが、センサ部をのぞき込み、目の奥まで光を当てられるという動作にユーザが抵抗感を感じたり、慣れが必要であることなどの制約がある。ハイエンドな入門管理用途に多い。
- 掌形 - 掌の形に加え指の関節間の長さや太さ、横方向から計った手の厚さなどを利用する。使いやすさもあって防衛関係の入門管理などでかなりの実績がある。
- 顔 - 常時露出しているため、抵抗感なく自然に入力できる。照合方式として、目や口などの代表点の位置をもとに構造をコード化して照合する、あるいは何らかの計測法を用いて鼻の高さや頬の形のような三次元構造を抽出し、それを用いてマッチングを行う、など様々な方法が研究されている。応用分野に応じて製品化が始まり、実用化フェーズに入りつつある。
- 音声 - 声紋に現れる個人性を利用する。周囲雑音、音声の品質、本人音声の変動、録音音声への対応など様々な技術的課題があるものの、心理的抵抗が少ない、特殊な入力装置が不要などの利点を持ち、電話を通じての情報サービスなどでの実用化が始まっている。
- 動的署名 - 特殊なペンタブレットを利用し、筆跡に加え筆順、運筆速度パターン、ペンをあげている間の軌跡、筆圧などの書き方の情報を個人の特徴とする方式が実用化されている。「練習」への対策などの課題があり精度は絶対的に高いとは言えないが、ユーザにとって受け入れやすいという利点があり、ペン方式の情報端末上で利用するにはコストの点からも適した方法と言える。
- 指紋 - 人間の指先の皮膚紋様である指紋は、「万人不動」、終生不変」という特徴を持つとされ、自動照合への取り組みも1960年代からと比較的長い歴史を持っている。

情報システムでの認証用途に一般の人々が使用することを考慮すると、「最高レベルの高精度照合方式がパーソナル分野においても利用可能である」、「入力センサが接触型で小型化できるために、パソコンや携帯端末に搭載するという形態でしやすい」という点で、低価格・小型な情報セキュリティを実現することに適していると考え

えられ、最も実用化が進んでいる<sup>(1) (9)</sup>。

これまでの個人認証には指紋を用いることが多かったが、最近では犯罪者の中で指紋をつぶす者があらわれたため、本研究では、より精度を高めるために指紋ではなく掌紋取り上げて個人認証を行うことにする。

認識にはパターン識別に高度な能力を発揮するニューラルネットワークを用いる。本研究では様々な条件で掌紋の認識を行い有用性について検証する。

## 2章 個人認証とは

ここでは、個人認証の活用場所について述べる。代表的なものとしては、銀行における個人認証、クレジットカードにおける個人認証、アクセス管理における個人認証、入退場システムにおける個人認証などがあげられる。個人認証には、様々な方法があるが、大きく3つに分類することができる。

バイオメトリクス、所有物情報、秘密情報があり、本研究では盗用する事が困難とされる、バイオメトリクスを取り上げる。

### 2.1 バイオメトリクスを用いた個人認証

バイオメトリクス（生物学的特徴）は、2つに分類され、本人の意思で変えることができない特徴点に注目するものと、恣意的に変えることができる特徴に注目するものがある。前者には、顔貌、網膜、虹彩、耳、指紋、掌紋、掌形、指形、手の甲の血管模様などが該当する。後者には、声紋、署名（筆跡）などが該当する<sup>(1)</sup>。

### 2.2 掌紋による個人認証

掌紋とは掌全体に見られる指紋様の皮膚隆起線のことであり、紋様は一生不変で遺伝性である。このため再現性が高いと考えられる。

また、掌紋画像の識別は、圧力と汗に対して強いと予想され、手の平の紋様と形状（掌形）とを利用することができると考えられる。

上記のように手の平の紋様には多くの情報が集約されていると考えられ、個人を識別するのに適した特徴と考えられる<sup>(2),(3)</sup>。

## 3章 ニューロ識別システム

本研究で採用するニューロ識別システムは、従来ニューラルネットワークによる紙幣識別処理において、学習・評価のシミュレーションを行うためのツールである。今回はこのツールを用いて掌紋画像の識別処理を行う。以下、ニューロ識別システムと記述する。

### 3.1 システムの主な機能

学習システムは主に以下のような機能をサポートする。

- ・ 掌紋画像データをブロック化するための情報ファイル（マスクファイル）を作成する。
- ・ 掌紋画像データとマスクファイルからニューラルネットワークの入力ファイル（スラブファイル）を作成する。
- ・ スラブファイルによりニューロ学習を実行し、ウェイトファイルを作成する。
- ・ ニューラルネットワークの入力値、出力値を評価するためのトラップファイルを作成する。
- ・ 上記ファイルを使用して掌紋画像ファイルを識別評価する。

### 3.2 ニューラルネットワーク

ニューラルネットワークの学習とは、与えられた入力に対して期待する出力が得られるようにネットワークの内部状態を調整することである。

学習の具体的な処理について（1 データ文）簡単に説明する。

- ・ 学習ファイルに含まれるスラブ値（入力値）をニューラルネットワークに入力する。
- ・ その時点におけるウェイト値によってニューラルネットワークの出力値を計算する。
- ・ 得られた出力値と教師値との誤差によってウェイト値を修正する。

以上の3点が学習の基本動作である。

### 3.2.1 ニューラルネットワーク

ニューラルネットワークは、図 3.1 に示すような3層の階層型ネットワークを使用している。各層のニューロン（細胞）の数は最大50個までとする。

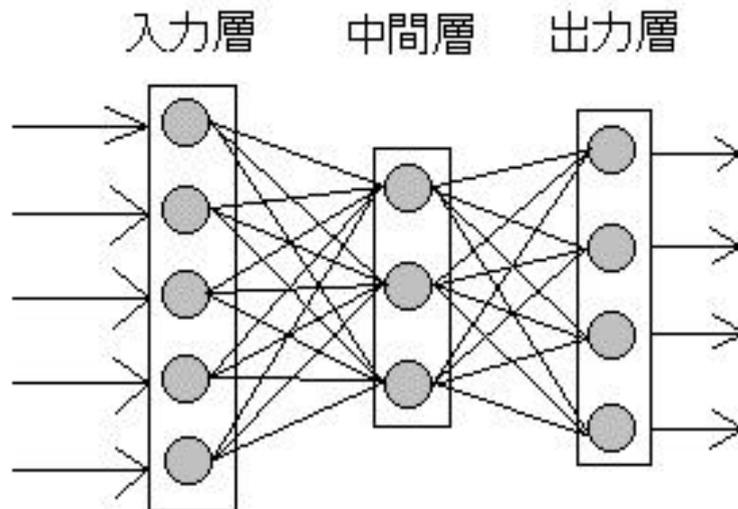


図 3.1 ニューラルネットワークの例

丸印で表されたのがニューロンで、各ニューロンは、入力層、中間層、出力層のそれぞれ独立した層に配置する。各層のニューロンはそれより1つ前の層のニューロンから入力を受ける。また、各層間ではすべてのニューロンが結びついている。

情報の流れる方向は、入力パターン（5つの入力値）が入力層の各ニューロンに与えられ、中層間で変換されて、出力層の各ニューロンから出力パターン（4つの出力値）が得られることになる<sup>(10)~(12)</sup>。

（用語）ニューロン：神経細胞。ユニットと同意で使用する。ネットワークのノードに当たる要素で、多入力、一出力の素子。

### 3.2.2 ネットワークの構成

ネットワークの構成は環境設定ファイルで設定する。設定項目は入力層の細胞数、中間層の数、中間層の細胞数、出力層の細胞数の4つである。

入力層の細胞数は入力スラブ値と同数に設定する。（マスクパターン）

中間層の数は現状では必ず“1”に設定すること。

中間層の細胞数は任意に設定できるが、入力層の細胞数と同数程度が適当である。

学習が収束しにくい時は中間層の細胞数を増やしてみるとよい。

出力層の細胞数は学習パターン数と同数に設定する。

各層とも、設定できる最大の細胞数は50である。

### 3.2.3 情報の伝達

ニューラルネットワークにおいて、入力層に与えられた入力パターンが、どのようにして伝達され、出力層から出力されるのかを説明する。

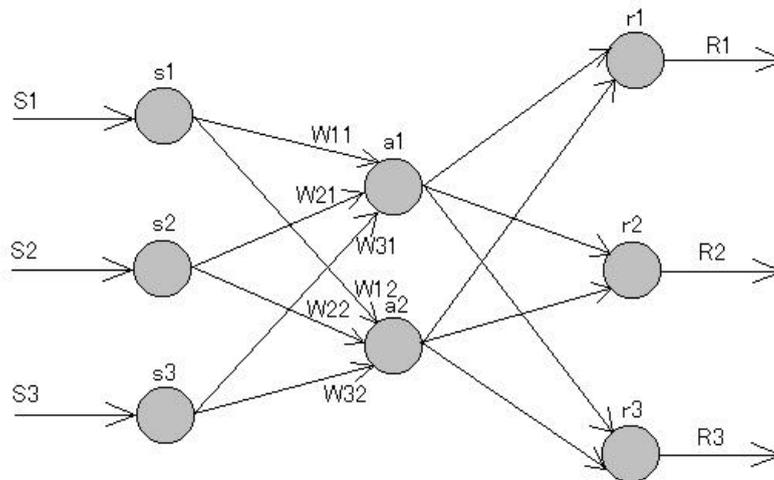


図 3.2 情報の伝達

図 3.2 で、 $s_1$ 、 $s_2$ 、 $s_3$  はそれぞれ入力層の第 1、2、3 ユニット、 $a_1$ 、 $a_2$  はそれぞれ中間層の第 1、2 ユニット、 $r_1$ 、 $r_2$ 、 $r_3$  はそれぞれ出力層の第 1、2、3 ユニットである。また、 $S_1$ 、 $S_2$ 、 $S_3$  は入力層への入力値（スラブ値）、 $R_1$ 、 $R_2$ 、 $R_3$  は出力層の出力値を表す。

入力層の第  $i$  ユニットから中間層の第  $j$  ユニットへの結合の重み（ウェイト）を  $W_{ij}$  とする。中間層と出力層の間の情報伝達方法もまったく同じであるから、入力層と中間層を例にして説明する。

#### <ユニットの入力値>

入力層の第  $i$  ユニットの出力値を  $S_i$  とすると（入力層のユニットは入力値と出力値が同じ）、中間層の第  $j$  ユニットの入力の総和  $I_j$  は式 3.1、式 3.2 をつかい、つぎのように求める。

$$\begin{aligned} I_1 &= W_{11}S_1 + W_{21}S_2 + W_{31}S_3 \\ I_2 &= W_{12}S_1 + W_{22}S_2 + W_{32}S_3 \end{aligned} \quad \dots (3.1)$$

となり一般的に、

$$I_j = \sum_i W_{ij} S_i \quad \dots (3.2)$$

と表すことができる。

< ユニットの出力値 >

中間層の第 j ユニットの総和を  $I_j$  とすると、中間層の第 j ユニットの出力値  $O_j$  は式 3.3 のようにして求める。

$$O_j = f(I_j) \quad \dots (3.3)$$

( f は入出力関数 )

このようにして、入力その各ユニットの出力値と、入力層と中間層の各ユニット間のウェイト間のから中間層の各ユニットの出力値を得ることができる。

同様にして、中間層の各ユニットの出力層と、中間層と出力層の各ユニット間のウェイトから出力層の各ユニットの出力値を得ることができる。

### 3 . 2 . 4 中間層・出力層の入出力関数

中間層・出力層の各ユニットの入出力関数としては、ロジスティック関数 ( シグモイド関数 )

$$f(x) = \frac{1}{1 + \exp\left(\frac{-x + \theta}{T}\right)} \quad \dots (3.4)$$

を使用する。

式 3.4 において、 $x$  は各ユニットへの入力値で、 $f(x)$  はそのユニットの出力値である。  $T$  はネットワークの温度と呼ばれる正の数で、 $T$  が大きくなるほどグラフはなだらかになる。  $\theta$  はユニット単位の閾値である。 図 3.3 にシグモイド関数グラフを示す (  $\theta = 0$  )

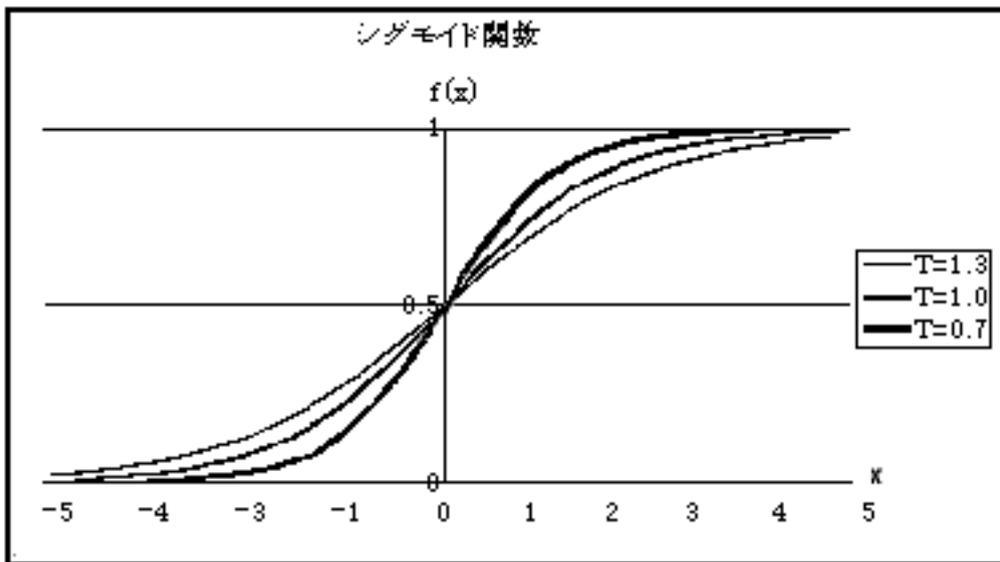


図 3.3 シグモイド関数

T(ネットワークの温度)は総合誤差に比例して、1.3 から 0.7 まで変化させる。この操作は学習プログラムが自動的に行う。また、総合誤差については後述する。

### 3.2.5 学習アルゴリズム

<(3.1.3) 情報の伝達>で、入力層に与えられた入力パターンが中間層を経て変換され、出力層から出力パターンが得られることを示したが、入力パターンに対して期待する出力パターンを得るためには、各ユニット間の結合重み(ウェイト)を適切な値にしなければならない。このウェイトを適切な値に設定するために学習を行う。

学習方法としては、誤差逆伝搬(Error Back Propagation)アルゴリズムを使用する。具体的には、いくつかの入力パターンの例(学習データ)を与え、その時の出力パターンと期待する出力パターン(教師値)との誤差が減少するようにウェイトを修正する。

ある入力パターンを与えた時の出力層の第jユニットの出力値を $O_j$ 、このときの出力層の第jユニットの期待値(教師値)を $T_j$ とすると、第jユニットの誤差 $E_j$ は式3.5のようにして求める。

$$E_j = \frac{1}{2}(T_j - O_j)^2 \quad \dots (3.5)$$

従って、1つの学習パターンPにおける出力層の誤差  $E_p$  は式 3.6 のようにして求める。

$$E_p = \frac{1}{2} \sum_j (T_j - O_j)^2 \quad \dots (3.6)$$

全学習パターンの誤差の総和を  $E$  とし、総合誤差と呼ぶことにする。

$$E = \sum_p E_p = \frac{1}{2} \sum_p \sum_j (T_j - O_j)^2 \quad \dots (3.7)$$

式 3.7 の総合誤差  $E$  が最小になるように各ユニット間の結合重み(ウェイト)を修正する。

修正の方法としては急降下法を用いる。具体的には、各学習パターン毎に、各ユニットの誤差  $E_j$  が最小になる方向へ微妙な変更を加えていく。

第  $t$  回目の学習における、 $k - 1$  層の第  $i$  ユニットから  $k$  層の第  $j$  ユニットへのウェイト  $W_{ij}$  の修正量  $\Delta W_{ij}^{k-1,k}(t)$  は以下のようにして求める。

$$\Delta W_{ij}^{k-1,k}(t) = -e d_j^k O_i^{k-1} + a \Delta W_{ij}^{k-1,k}(t-1) + b \Delta W_{ij}^{k-1,k}(t-2) \quad \dots (3.8)$$

式 3.8 において、“ $e$ ” は学習定数、“ $a$ ” は慣性定数、“ $b$ ” は振動定数である。

また、“ $d_j^k$ ” は  $k$  層の第  $j$  ユニットの一般化誤差で、 $k$  層が出力層の場合と中間層の場合によって算出方法が異なる。以下に一般化誤差の算出方法を示す。

$k$  層が出力層の場合、( $I_j^k$  は  $k$  層の第  $j$  ユニットの入力総和)

$$d_j^k = (T_j - O_j^k) f'(I_j^k) \quad \dots (3.9)$$

$k$  層が中間層の場合、(ただし、 $m$  は出力層のユニット番号)

$$d_j^k = \left( \sum_m W_{jm}^{k,k+1} d_m^{k+1} \right) f'(I_j^k) \quad \dots (3.10)$$

以上が誤差逆伝搬法によるウェイト修正の概略である。  
詳しくは専門書を参考にして頂きたい。

### 3.2.6 学習パラメータ

前項で示したウェイトの修正式

$$\Delta W_{ij}^{k-1,k}(t) = -e d_j^k O_i^{k-1} + a \Delta W_{ij}^{k-1,k}(t-1) + b \Delta W_{ij}^{k-1,k}(t-2) \quad \dots (3.11)$$

式 3.11 における “ ” 学習定数、“ ” 慣性定数、“ ” 振動定数は環境設定ファイルで設定する。

“ ” 学習定数は大きな値にするとウェイトの修正量が大きくなり、学習速度は速くなるが、あまり大きくしすぎると逆に学習が収束しなくなる。総合誤差が上下に振動するときは学習定数を小さくし、誤差の減少速度が小さいときは学習定数を大きくする必要がある。この操作は学習プログラムが自動的に行う。ユーザーは学習開始時の学習定数の初期値を設定する。初期値は  $(0.1 < < 1.0)$  の範囲で設定すること。デフォルト値は  $0.5$  である (図 3.4)。

- “ ” 慣性定数は総合誤差の振動を減らし、学習の収束を加速させる働きをする。
  - “ ” 振動定数は総合誤差を上下に振動させて極小値から脱出させる働きをする。
- と には関連性があり、以下の範囲内 (塗りつぶした部分) で設定すること。

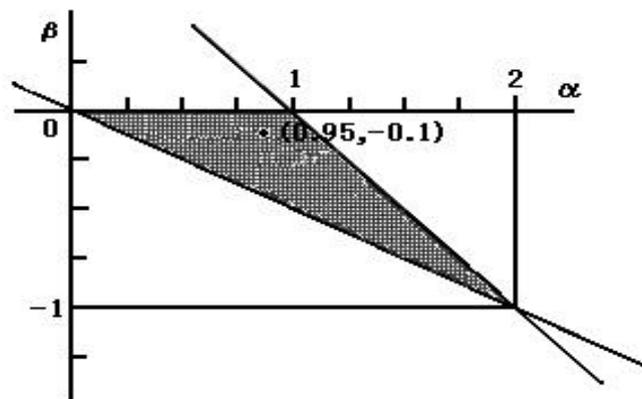


図 3.4 学習定数の範囲

慣性定数のデフォルト値は0.95、振動定数のデフォルト値は-0.1である。

### 3.3 学習システムの環境

学習を行う対象、目的に応じてシステムの環境を設定する必要がある。

#### 3.3.1 データベース設定ファイル

データベース設定ファイル“DATABASE.CFG”(テキスト形式)を作成する。データベースファイルのパス名をパターン番号順に1行に1ファイル記述する。以下に、例を表示する(図3.5)。

```
File name: E:\NNC最~1\DATABASE.CFG
e:\scandemo\data01.db<
e:\scandemo\data02.db<
e:\scandemo\data03.db<
e:\scandemo\data04.db<
e:\scandemo\data05.db<
e:\scandemo\data06.db<
```

図3.5 DATABASE.CFG

学習データはこのデータベース設定ファイルに基づいて抽出されるので、記述する順番は重要である。

#### 3.3.2 環境設定ファイル

環境設定ファイル“NEURO.CFG”(テキスト形式)を作成する。ニューラルネットワークの構成、学習パターン数など11項目を記述する。設定する項目と設定値の範囲は表3.1の通りである。

表 3.1 NEURO.CFG の設定

行	項目	設定値(x)の範囲
1	入力層の細胞数	2 x 50(整数)
2	中間層の数	x=1(整数)
3	中間層の細胞数	2 x 50(整数)
4	出力層の細胞数	2 x 50(整数)
5	学習定数の初期値	x=0.05
6	慣性定数	x=0.95
7	振動定数	x=-0.1
8	学習パターン数	2 x 50(整数)
9	1パターンの学習枚数	2 x 65534(整数)
10	最終誤差判定値	0.0<x
11	最大学習回数	2 x 65534(整数)

上記の 11 項目について数値とコメントを記述する。ファイルはテキスト形式である。1 行に 1 項目とし、数値の右側にスペースまたはタブで区切ってコメントを書く。コメントは半角 32 文字（全角 16 文字）までとし、省略はできない。また、コメントにスペース、タブを使用してはならない。

以下に、例を示す（図 3.6）。（5 パターンで各 2 枚学習する場合）

```
File name: E:\NNC最~1\NEURO.CFG
50      入力層細胞数↵
1       中間層数↵
30      中間層細胞数↵
5       出力層細胞数↵
0.050000 学習定数↵
0.950000 慣性定数↵
-0.100000 振動定数↵
5       学習パターン数↵
2       1パターンの提示枚数↵
0.000100 最終誤差判定値↵
20000   最大学習回数↵
```

図 3.6 NEURO.CFG

各項目を設定する時は、以下のことに注意する。

- ・ 入力層の細胞数はスラブ値の個数と等しくする（ マスクファイル作成参照 ）。
- ・ 中間層の細胞数は入力層の細胞数と同数程度にする。
- ・ 学習パターン数は学習するデータベース数と同じである。
- ・ 1 パターンの学習枚数はデータベースファイルの枚数以下にする必要がある。
- ・ 最終誤差判定値と最大学習回数は目的によって設定値が異なるが、通常は上記の例と同じ値にすること。

### 3.4 学習システムの動作

ここでは、システムとしての動作及び各プログラムの詳細な動作説明をする。

#### 3.4.1 学習データ抽出 / 追加

第 1 回目の学習時は、データベースから学習に使用するデータを抽出する。抽出の方法は、環境設定ファイルの “ 1 パターンの学習枚数 ” 文だけ各データベースファイルの先頭から採り出す。このようにして、学習用のデータファイルを作成する。 EX\_LDATA。 EXE

ファイル名は “ LDATA \* ” となる。（ \* はパターン番号 ）

#### 3.4.2 マスクファイル作成

ニューラルネットワークの入力値として画像データを使用する場合、複数の画素の有効 / 無効を選択する必要がある。この有効 / 無効（スリット / マスク）をあらわすデータのことをマスクデータと呼ぶ。マスクデータはスラブ値を計算する時に使用する。

データベースファイルにスラブ値が格納されていて、学習、評価共にそのスラブ値を使用する場合は、マスクファイルは必要ない。マスクファイルを使用するかしないかは、スラブ値作成（ MKSLAB . EXE ） 評価（ EVAL50 . EXE ） を実行する時にモードを指定することができる。

データベースファイルにスラブ値が格納されていない場合や、マスクパターンの検討段階においてはマスクファイルを作成する必要がある。

MASK.EXEはMASK.TXTに基づいてマスクファイル(MASK.M)を作成する。

以下、マスクファイルを使用する場合に、その作成の方法について説明する。

### (1) マスク単位と領域を決定する

最初にマスク単位の大きさや数(領域全体)を決定する。

例として、図3.7のようなマスクを考える。

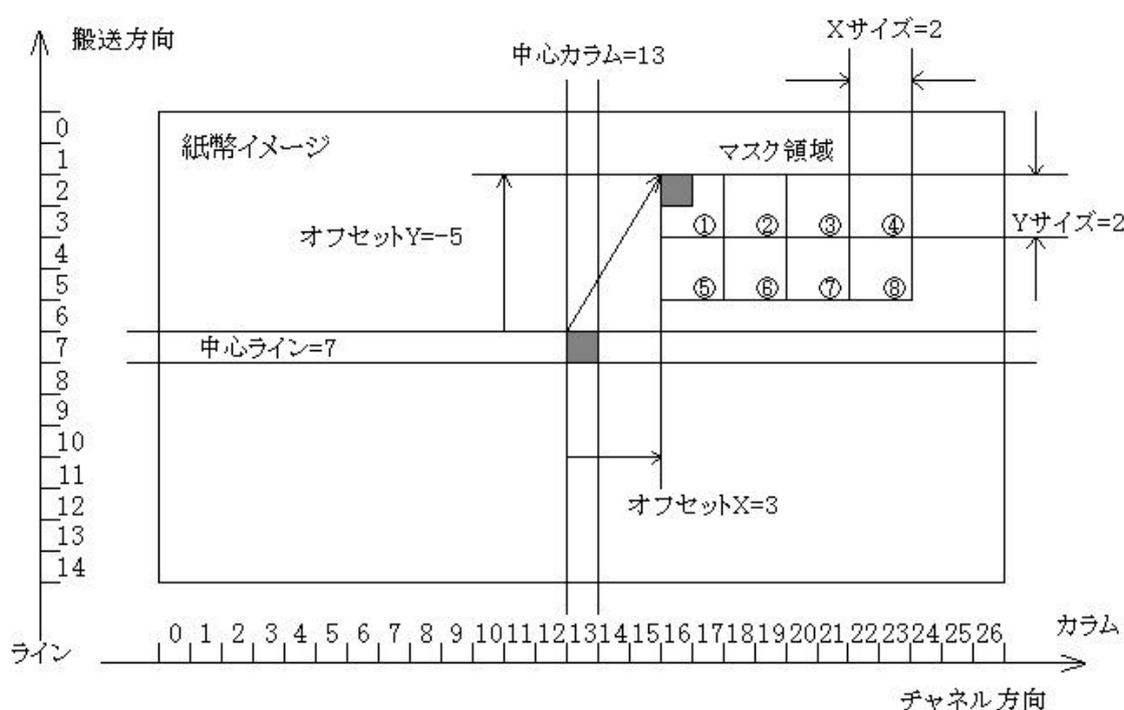


図 3.7 マスク領域

- ・ 紙幣イメージの最初(左上)の画素が座標(0, 0)にあるものとする
- ・ マスク単位は、Xサイズ=2、Yサイズ=2で4画素
- ・ マスク領域におけるマスク単位の数は、X方向=4個、Y方向=2個で合計8個
- ・ マスク領域全体は、8(チャンネル) \* 4(ライン) = 32画素
- ・ 中心からのオフセットは、(3, -5)

マスク領域(全体)の位置を示すために、中心座標に対するマスク領域の最初(左上)の画素のオフセットを決定する。

中心画素の座標は次のようにして求める。

$$\text{中心画素の X 座標} = \text{紙幣の X 方向サイズ} / 2 = 27 / 2 = 13$$

$$\text{中心画素の Y 座標} = \text{紙幣の Y 方向サイズ} / 2 = 15 / 2 = 7$$

オフセットは次のようにして求める。

マスク領域の第1マスク単位 の第1画素の座標が (16, 2) であるから、

$$\text{オフセット (X, Y)} = (16, 2) - (13, 7) = (3, -5)$$

となる。

### (2) マスクパターンを決定する

つぎに、マスク領域の全マスク単位についてスリット/マスクのパターンを決定する。

例として、図 3.8 のようなマスクを考える。(前項と同じ)

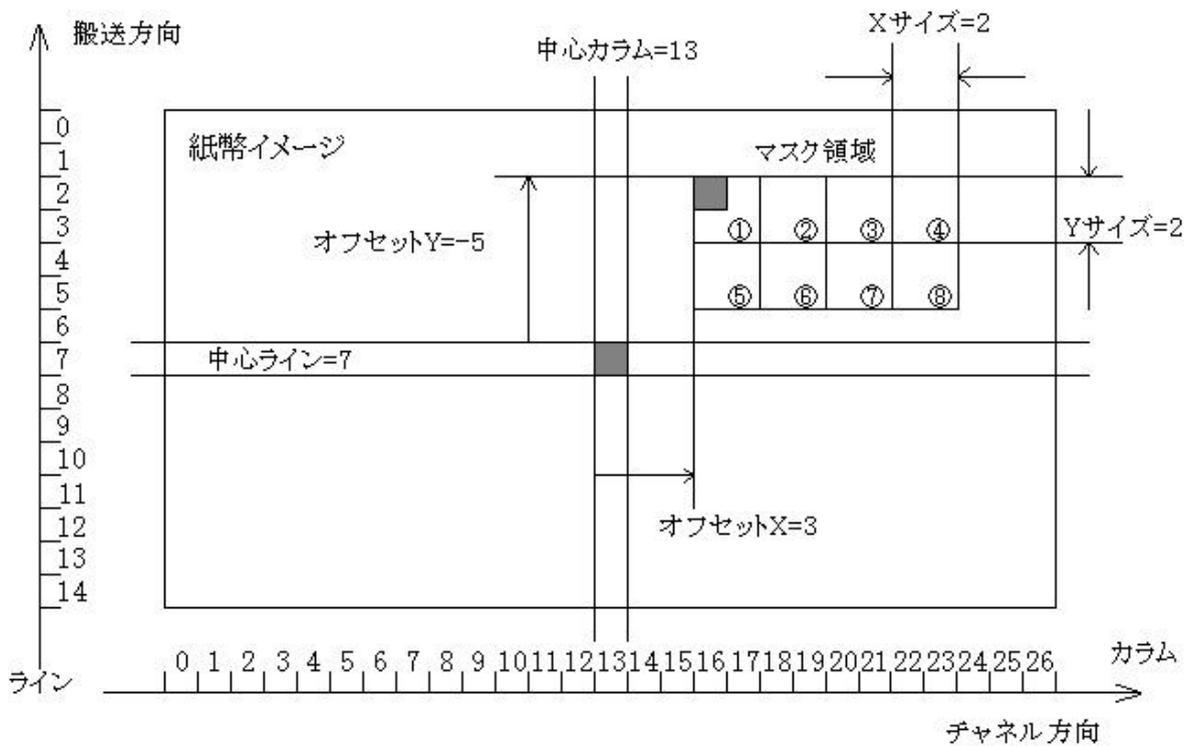


図 3.8 マスクパターン

図 3.8 ではマスク単位が 8 個有り ( ~ )、1 つのマスクパターンは 8 個のデータで表される。1 つのデータは “ 0 ” か “ 1 ” (マスクかスリット) で表す。

例えば、図のように をスリットとし、 ~ をマスクとするならば、そのマスクパターンは

“ 1 0 0 0 0 0 0 0 ” となる。データの並びは ... の順である。同じように、 ~ をすべてスリットとするなら、マスクパターンは “ 1 1 1 1 1 1 1 1 ” となる。

### ( 3 ) マスク設定ファイルを作成する

マスクの方法が決定したら、それをテキストファイルに記述する。

テキストエディタを用いてファイル “ MASK . TXT ” を作成する ( 図 3.9 )。

```
File name: E:\¥NNC最~1¥MASK.TXT
1      マスクの i d 番号↵
-60    中心からのオフセット X↵
-15    中心からのオフセット Y↵
12     マスク単位の X サイズ↵
5      マスク単位の Y サイズ↵
12     マスク単位の数 ( X 方向 ) ↵
4      マスク単位の数 ( Y 方向 ) ↵
50     マスクパターンの数↵
——MaskData——↵
```

図 3.9 MASK.TXT

このファイルは MASK . EXE を実行する際に必要である。

## 3 . 4 . 3 スラブファイル作成

MKSLAB . EXE を実行すると、スラブファイル ( SLAB\* ) を出力する。

モード 0 で実行すると、学習用のデータファイルからスラブ値だけを採りだします。学習用のデータファイルにスラブ値が含まれていない場合はモード 0 で実行することはできない。

モード 1 か 2 で実行すると、マスクファイル ( MASK . M ) に基づいて、イメージデータからスラブ値を計算する。

以下、スラブ値の求めかたと出力ファイルのフォーマットについて記述する。

( 1 ) スラブ値の求め方

マスクファイル ( M A S K . M ) に基づいて、有効であるマスク単位の画素値の総和を求める。これをマスクパターン数分だけ求め、マスクパターン ( i ) の総和を  $s(i)$ 、有効な画素数を  $n(i)$  とする。

マスク領域全体の中で画素の最大値、最小値を求め、それぞれ  $max$  ,  $min$  とする。

このとき、第  $i$  スラブ値 :  $slab(i)$  は式 3.12 のように求める。

$$slab(i) = \frac{1}{max - min} \left( \frac{s(i)}{n(i)} - min \right) \quad \dots (3.12)$$

つぎに第 1 スラブ値  $slab(1)$  でその他のスラブ値を正規化する。

従って、第 1 スラブ値は特別な意味を持つことになる。

また、第 1 スラブ値は他のスラブ値とレンジを合わせるために 2 倍にする。

最終的に第  $i$  スラブ値 :  $SLAB(i)$  は式 3.13、式 3.14 を用いてつぎのようにして求める。

$$SLAB(1) = slab(1) \times 2 \quad (i \neq 1) \quad \dots (3.13)$$

$$SLAB(i) = \frac{slab(i)}{slab(1)} \quad \dots (3.14)$$

第  $i$  スラブ値はニューラルネットワークの入力層の第  $i$  ユニットへ入力される。  
詳しくは学習の章を参照すること。

( 2 ) スラブファイルのフォーマット

スラブファイルはパターン数分だけ出力される。名前は  $SLAB*$  ( \* はパターン番号 )。

1 枚分のフォーマットは以下の通りで、ヘッダ部分 8 バイトとスラブ値データで構成される。各々のデータはすべてバイナリ形式である。

表 3.2 スラブファイルのフォーマット（1枚分）

+0	パターン番号	(1バイト)
+1	金種情報	(1バイト)
+2	方向情報	(1バイト)
+3	国情報	(1バイト)
+4	種別	(1バイト)
+5	予約領域	(1バイト)
+6	通し番号	(2バイト)
+8	第1スラブ値	(4バイト)
+12	第2スラブ値	(4バイト)
	第3スラブ値	(4バイト)
⋮	⋮	
	第nスラブ値	(4バイト)

- ・ヘッダ部分の各項目はデータベースファイルの内容と同じである。
- ・スラブ値はIEEE浮動小数点フォーマットである。

### 3.4.4 学習ファイル作成

各スラブファイルから1枚分のスラブ値データ（ヘッダを含む）を順次採りだし、パターン順に並べて学習ファイルを作成する。学習ファイルのフォーマットはスラブファイルと同じである。

図 3.10 に処理のイメージを示す。

### 3.4.5 教師ファイル作成

各学習パターンに対応する出力層の出力値（期待する出力値）を記述したテキストファイル“TEACH.T”を作成する。図 3.11 に教師ファイルの例を示す。

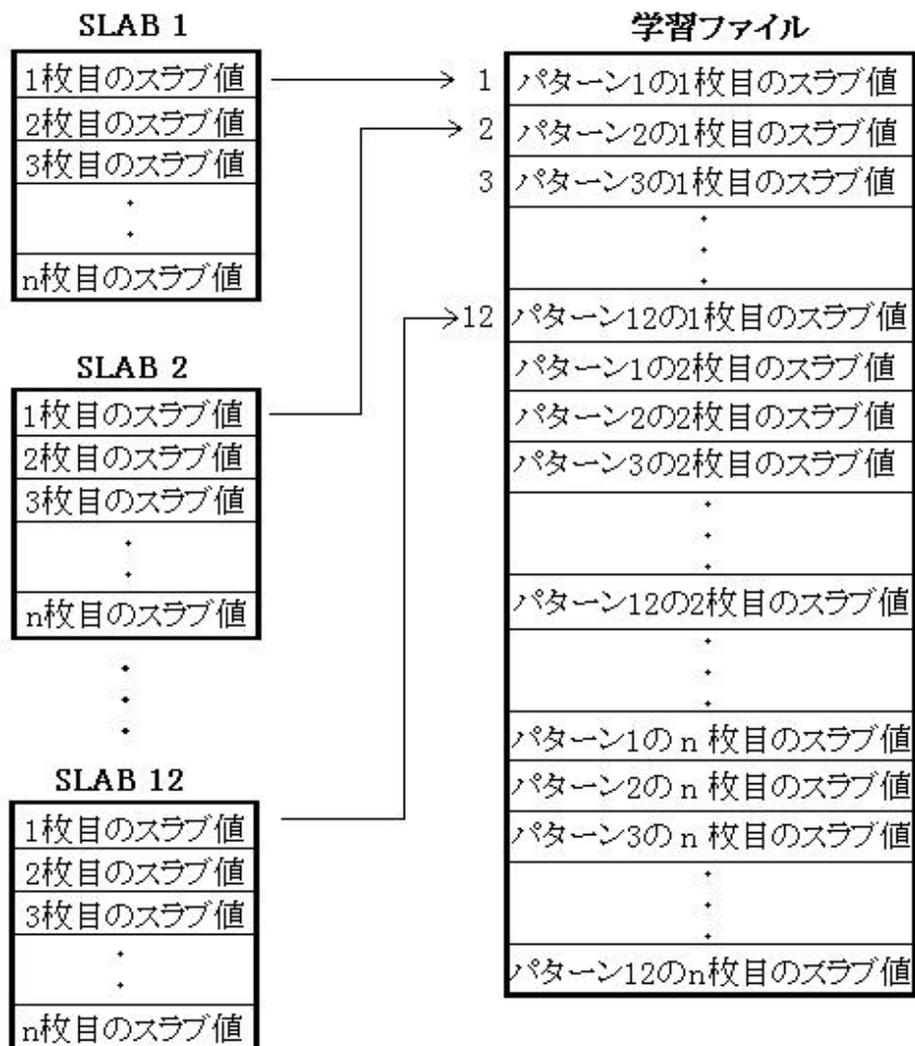


図 3.10 学習ファイルと各スラブファイルの関係

```
File name: E:\NNC最~1\TEACH.T
1 0 0 0 0 0 0 0 ←
0 1 0 0 0 0 0 0 ←
0 0 1 0 0 0 0 0 ←
0 0 0 1 0 0 0 0 ←
0 0 0 0 1 0 0 0 ←
0 0 0 0 0 1 0 0 ←
0 0 0 0 0 0 1 0 ←
0 0 0 0 0 0 0 1 ←
```

図 3.11 教師値例

図 3.11 の数値部分が教師ファイル（テキスト形式）の内容である。

1 パターンの教師データを 1 行に記述する。この場合出力層が 8 ユニットなので、8 個の数値（教師値）になる。各値はスペースまたはタブで区切る。8 パターンなので 8 行セットする。1 行目はパターン 1 の教師データであり、出力層の第 1 ユニットに “ 1 ” を、それ以外のユニットには “ 0 ” を出力させることを示している。同じように、2 行目はパターン 2 の教師データであり、出力層の第 2 ユニットに “ 1 ” を、それ以外のユニットには “ 0 ” を出力させることを示している。他も同様である。

## 4章 ニューロ識別システムを用いた実験

### 4.1 実験目的

パターン識別に高度な能力を発揮するニューラルネットワークを用いた掌紋画像識別の有用性について検証する。

### 4.2 実験機器

実験に使用する機器はPC (Personal Computer)、イメージスキャナ、それとPCに組み込まれたニューロ識別システムである。

PCについては、研究室内のPCを使用する。

スキャナは一般に市販されているイメージスキャナを使用する。だが、画像の採取を行うときに条件に差が出ては公平な学習が行われず、正確な学習、認識ができないため採取条件をなるべくそろえる必要がある。そこで既存のスキャナでは採取域が大きく、手置き方にも差が出てくることに着目し、あらかじめ一定の範囲に決まったかたちに手を置けるようスキャナに工夫する。



図 4.1 イメージスキャナ

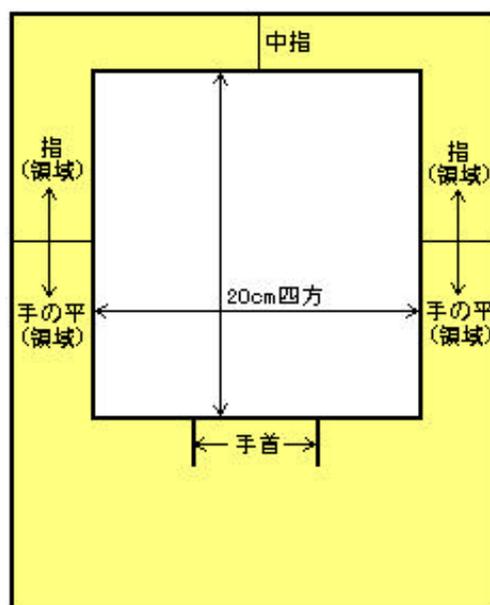


図 4.2 ダンボール製の型

ここで、あらかじめ手を置く位置が一定の範囲に収まるようにダンボール紙で型を作りスキャナに被せた。(図 4.1)このダンボールの型はスキャナに合わせた大きさにダンボールを切り、そこから 20cm 四方の範囲で切り抜いたものである。20cm 四方という数字は被験者全員の手の大きさから図ったものである。また、中指の来る位置と向き、それと手のひらの大体の位置を指定できるよう型に印をつけた。仕様は図 4.2 の通りである。

ニューロ識別システムについては 3 章を参照。

## 4.3 実験手順

### 4.3.1 実験の流れ

実験の流れは画像採取、学習、評価の順に行う。

画像の採取は基本的に 1 度行えばよい。採取した掌紋画像データは学習・評価に何度も使用できる。ただし、採取したままのファイル形式ではニューロ識別システムが受け付けないためニューロフォーマットを行う必要がある。

学習については学習枚数を変化させ比較することも考えているので何度か行う。

学習を行えば必然的に評価も行うことになるので評価も何度か行うことになる。

### 4.3.2 画像採取

はじめに人数についてだが、著者の研究室の協力者から 7 人協力してもらい、自分自身の掌紋もあわせ、計 8 人分の掌紋を採取する。それぞれ左手の掌紋をイメージスキャナを使用し採取する。

画像ファイルの形式についてはビットマップ形式で保存する。画像の大きさは縦 19.51cm (553 ピクセル) 横 18.17cm (514 ピクセル) である。

画像は解像度を 72 DPI にし、グレースケールで読み込む。

画像 1 枚あたりのサイズは約 300KB である。図 4.3 は採取した画像の一部である。

採取する画像の種類は中指が上を向いた状態を通常状態とし、各人 10 枚ずつ採取する(図 4.4)。つぎに図 4.5 にあるように中指を右斜め 30 度に向けて、5 枚ずつ採取する。その後、さらに 30 度右に回転させ合計 60 度回転させた状態で 5 枚採取する(図 4.6)。同じように左 30 度回転させた状態で 5 枚採取する(図 4.7)。最後に左に 60 度回転させた状態で 5 枚採取する(図 4.8)。

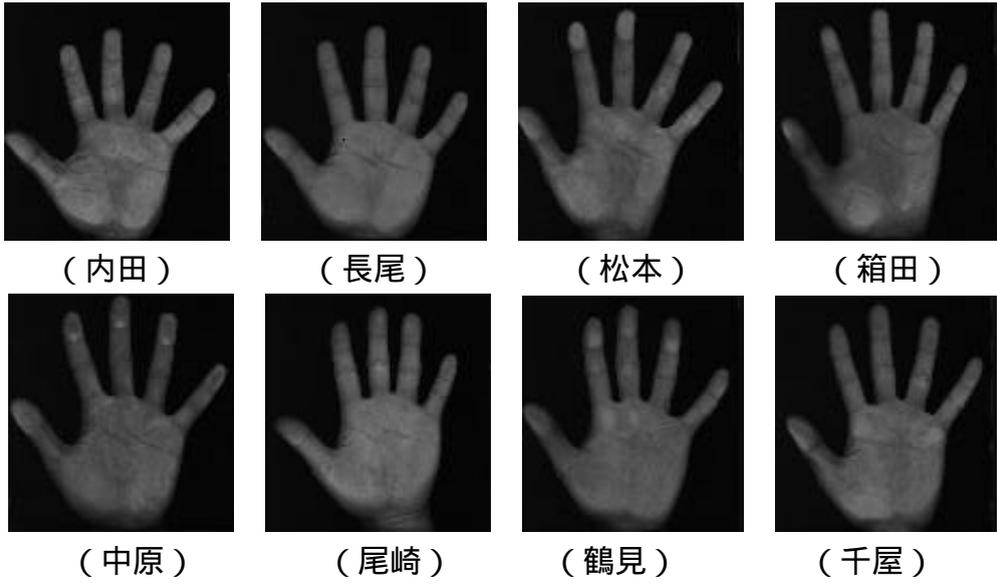


図 4.3 採取した画像の一部



図 4.4 通常状態



図 4.5 右に 30 度



図 4.6 右に 60 度



図 4.7 左に 30 度



図 4.8 左に 60 度

一人あたり 30 枚で合計 240 枚の画像データを採取する。合計のサイズは約 65 MB である。

その後、採取した掌紋画像をビットマップ形式の画像ファイルからニューロフォーマットである DB ファイルに変換しなければならない。ここで PC のモニタプログラムである scandemo の説明をする。まず、scandemo.cfg というファイル設定について説明する。図 4.9 ここで重要なのは切りだし領域の設定とパスである。実験で使用する掌紋画像は手全体の画像であるため画像切り出し開始行、画像切り出し開始列共に 0 とする。画像切り出し終了行を 550 とし、画像終了列を 520 とする。

```
File name: E:\$SCANDEMO\SCANDEMO.CFG

%SCANDEMO.EXE設定ファイル
%コメント部には空白を入れないこと
%コメントは3行
[NN_ID1(10進値)]      11
[NN_ID2(10進値)]      22
[NN_ID3(10進値)]      33
[NN_ID4(10進値)]      44
[層数]                 3
[学習パターン数&出力ユニット数] 8
[提示枚数]             10
[マスクID]             1
[入力ユニット数]       50
[中間ユニット数]       35
[目標誤差]             0.0001
[最大学習回数]         20000
[学習定数]             0.05
[慣性定数]             0.95
[振動定数]            -0.1
[温度勾配]            1.3
[入力データ種別(画像有/無)] 0
[入力スイッチ(別紙参照)] 1
[出力スイッチ(別紙参照)] 1

[学習モード(初期/継続)] 1
[表示インターバル]      2
[MKSLAB起動スイッチ]    1
[RT_SLAB起動スイッチ]   1
[書き込み条件1]         100
[書き込み条件2]         10
[表示条件1]              0.0005
[表示条件2]              0.001
[表示条件3]              0.005
[表示条件4]              0.01
[表示条件5]              0.05
[表示条件6]              0.1
[表示条件7]              0.5
[表示条件8]              1.0
[画像切り出し開始行]    0
[画像切り出し終了行]    550
[画像切り出し開始列]    0
[画像切り出し終了列]    520
[BMP_PATH]               .\hand\hand
%[フォントマップ作成用] C:\win\fonts\*.FON
```

図 4.9 scandemo.cfg

scandemo.exe を offline モードで実行すると図 4.10 の画面になる。ここで「2」の DATA CONVERT を選択するとビットマップデータが DB ファイルに変換される。(図 4.11)

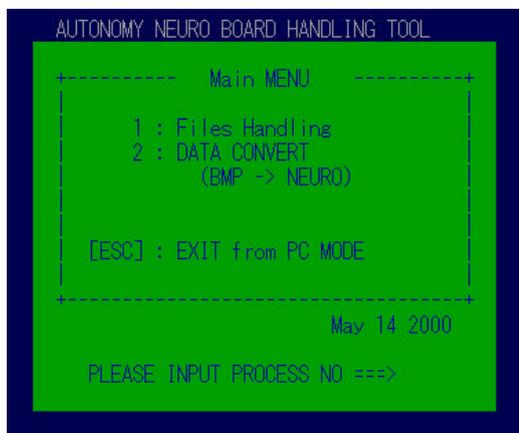


図 4.10 実行画面

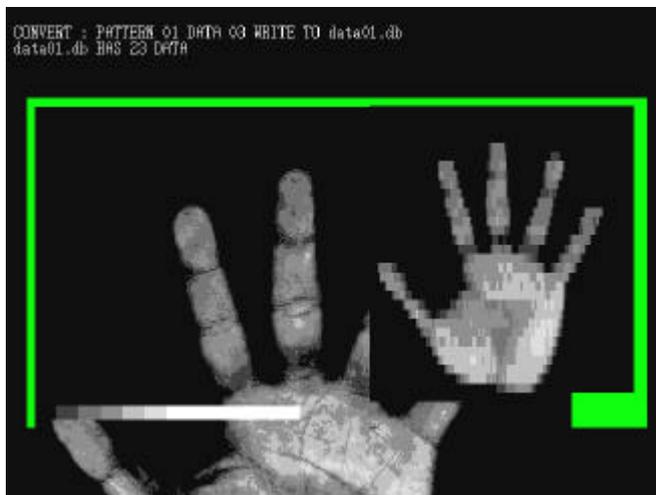


図 4.11 変換画面

### 4.3.3 学習

画像の変換が終了すると、学習である。ニューロ識別システムに画像データを学習させるにはまずデータベース設定ファイルである DATABASE.CFG (図 4.12) と環境設定ファイルである NEURO.CFG (図 4.13) の設定を行わなければならない。

図 4.12 の DATABASE.CFG の設定は 8 パターンの場合である。また、図 4.13 の NEURO.CFG の設定は 8 パターンの画像から 10 枚ずつ学習する場合の設定である。



図 4.12 DATABASE.CFG

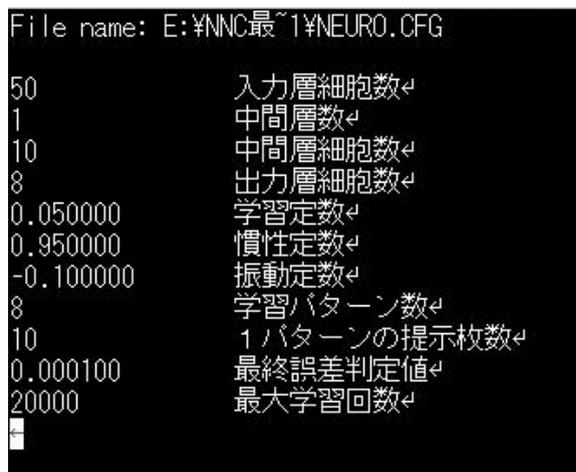


図 4.13 NEURO.CFG

DATABASE.CFG と NEURO.CFG の設定が終われば学習データの抽出に移る。ここでは環境設定ファイルで指定した 1 パターンの学習枚数文だけデータベース設定ファイルに登録されたデータファイルの先頭から取り出し、学習データファイル (LDATA 1 ~ 8 ) を作成する ( 図 4.14 )。

つぎにマスクの設定を行う。次は mask.txt のマスクの中心からのオフセット X、中心からのオフセット Y、マスク単位の X サイズ、マスク単位の Y サイズ、マスク単位の数(X 方向)、マスク単位の数(Y 方向)を図 4.15 に示すようにマスク設定を変更する。この設定でマスクをかけると図 4.16 のようになる。

つぎにスラブ値ファイルの作成を行う。スラブ値ファイルは学習データファイルのスラブ値を取り出して作成する。スラブファイル ( SLAB1 ~ 50 ) の作成中の画面を図 4.17 に示す。また、学習収束を早めるために、各パターンのスラブファイル( SLAB 1 ~ 5 0 ) を並び替え、学習ファイル ( LEARN.S ) を作成する。学習ファイルと各スラブファイルの関係は図 4.17 に示す。図中の n は 1 パターンの学習枚数である。



図 4.14 学習データ抽出

```

File name: E:\NNC最1\MASK.TXT
1      マスクのid番号↵
-100   中心からのオフセットX↵
-16    中心からのオフセットY↵
16     マスク単位のXサイズ↵
8      マスク単位のYサイズ↵
12     マスク単位の数 (X方向) ↵
4      マスク単位の数 (Y方向) ↵
50     マスクパターンの数↵
MaskData↵

```

図 4.15 MASK.TXT

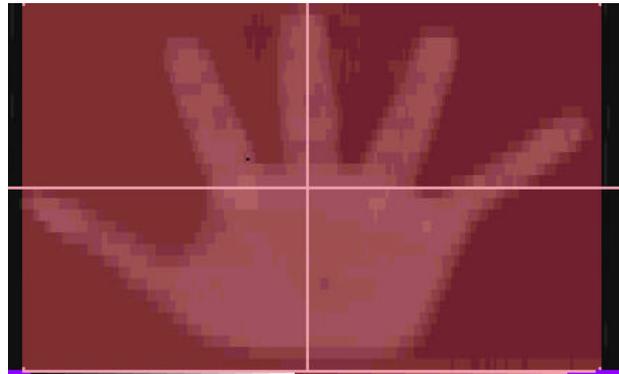


図 4.16 マスク領域

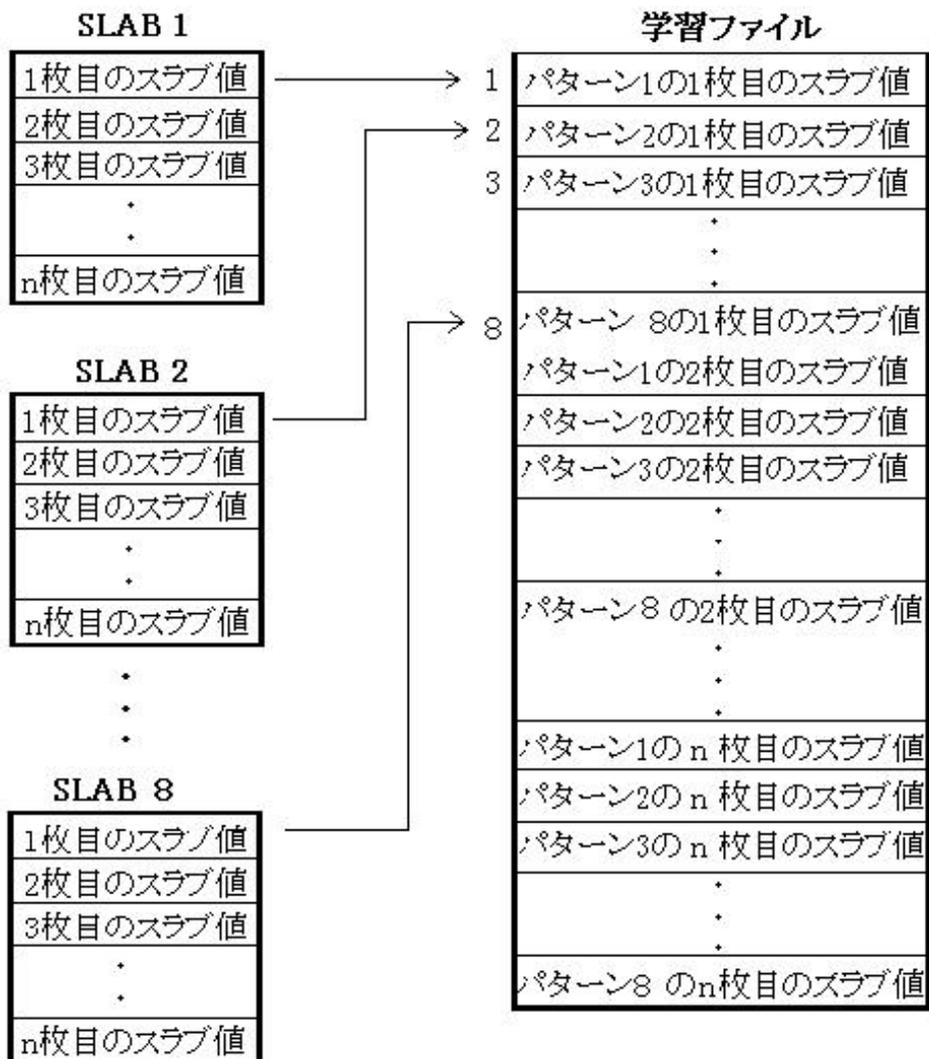


図 4.17 学習ファイルと各スラブファイルの関係

その後、学習を実行し、ウエイトファイルを作成する。学習プログラムは環境設定ファイルの最終誤差判定値または学習回数のどちらかの条件を満たしたときに終了する。図 4.18 中の左下は相互誤差を表し、右半分はそれぞれのパターンの出力ユニット値を表している。

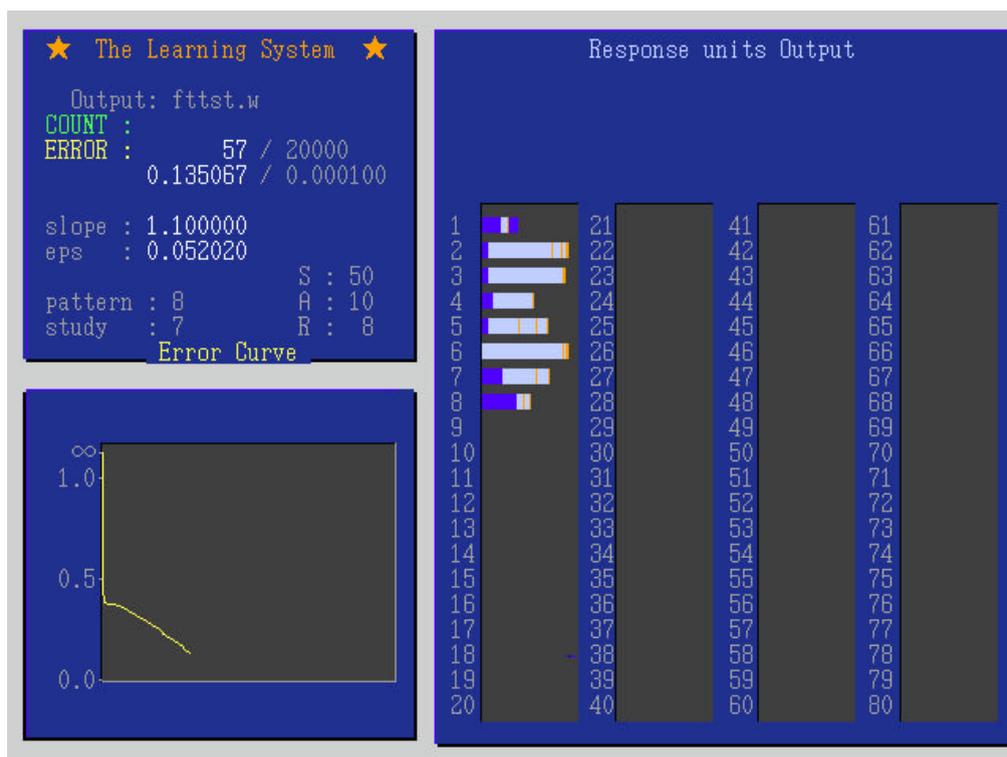


図 4.18 学習画面

#### 4.3.4 評価

ウエイトファイルを用いてデータファイルを評価し、評価結果としてログファイルを出力する。図 4.19 の評価実行時の画面中の右半分は 1 行目がパターン 1 の反応値、2 行目がパターン 2 の反応値を表す。ここではパターン 2 の画像を評価した場合にパターン 2 に一番強く反応したことを表している。

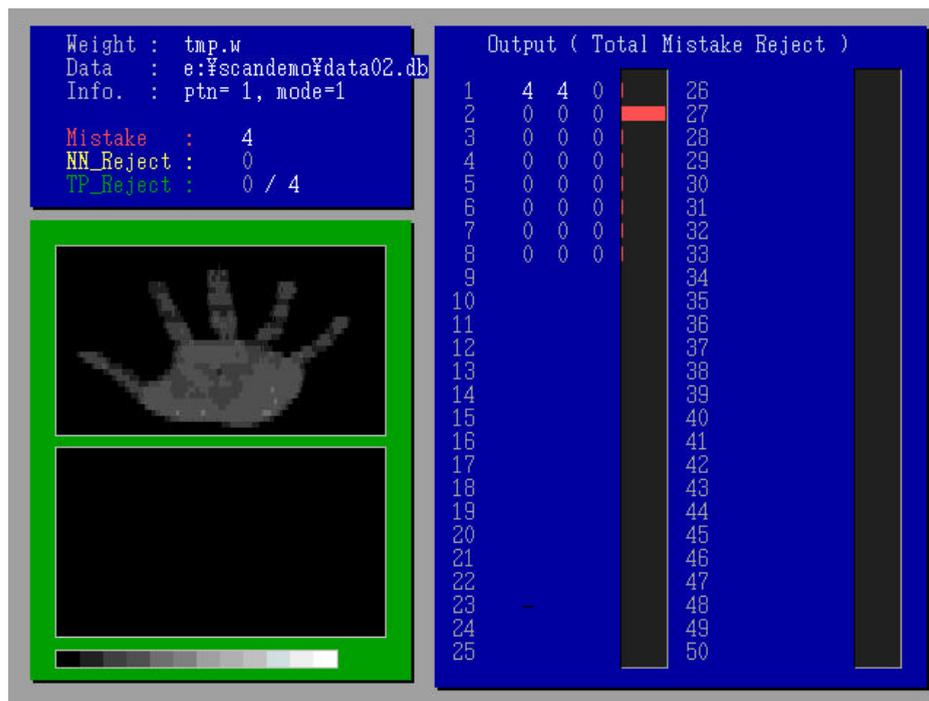


図 4.19 認証画面

## 4.4 実験条件

採取した画像を使用し、学習条件に変化を与えて認証率を比較する。ここでは4つの条件を与えて実験を行う。最初の実験で使用する画像は8パターンの通常画像10枚で、計80枚である。この内1パターン2枚ずつ学習させ識別する。続けて条件2として5枚学習、条件3として7枚学習と学習枚数に変化をつけ認証する。

その後、回転させた画像も使用し実験する。このときの学習条件は1パターンあたり30枚ある画像の内、通常画像2枚、回転させた画像4種類を2枚ずつ学習させるものとする。これを条件4とする。下記の表4.1は実験条件をまとめたものである。

(表 4.1 実験条件)

	1パターンあたりの学習枚数	全学習枚数	全識別枚数
条件1	通常画像2枚	16枚	80枚
条件2	通常状態5枚	40枚	80枚
条件3	通常状態7枚	56枚	80枚
条件4	5種類の画像を2枚づつ	80枚	240枚

## 4.5 実験結果

4章の条件で行った実験の結果を表5.1に示す。

認証率 = 正しく認証された枚数 / 全識別枚数 × 100 (%)とする。

(表 5.1 全体の認証率)

	1パターンあたりの学習枚数	正しく認証した枚数	全識別枚数	認証率
条件1	通常画像2枚	58枚	80枚	72.5%
条件2	通常状態5枚	77枚	80枚	96.3%
条件3	通常状態7枚	80枚	80枚	100.0%
条件4	5種類を2枚ずつ	209枚	240枚	87.1%

(表 5.2 条件1の認証率)

	正しく認証した枚数(10枚中)	認証率
パターン1	3枚	30%
パターン2	10枚	100%
パターン3	9枚	90%
パターン4	6枚	60%
パターン5	8枚	80%
パターン6	10枚	100%
パターン7	10枚	100%
パターン8	2枚	20%
合計	58枚	72.5%

(表 5.3 条件2の認証率)

	正しく認証した枚数(10枚中)	認証率
パターン1	9枚	90%
パターン2	10枚	100%
パターン3	10枚	100%
パターン4	10枚	100%
パターン5	9枚	90%
パターン6	10枚	100%
パターン7	10枚	100%
パターン8	9枚	90%
合計	77枚	96.3%

(表 5.4 条件 3 の認証率)

	正しく認証した枚数(10枚中)	認証率
パターン1	10枚	100%
パターン2	10枚	100%
パターン3	10枚	100%
パターン4	10枚	100%
パターン5	10枚	100%
パターン6	10枚	100%
パターン7	10枚	100%
パターン8	10枚	100%
合計	100枚	100%

(表 5.5 条件 4 のの認証率)

	正しく認証した枚数(30枚中)	認証率
パターン1	22枚	73.3%
パターン2	29枚	96.6%
パターン3	28枚	93.3%
パターン4	24枚	80%
パターン5	23枚	76.6%
パターン6	30枚	100%
パターン7	30枚	100%
パターン8	23枚	76.6%
合計	209枚	87.1%

上記の条件で実験を行った結果、学習枚数が多いほど認証率の向上が見られた。

## 5章 まとめ

実験結果からニューロ識別システムによる掌紋画像の識別は有効であることが判明した。また、学習枚数を多くすることで認証率が向上することが確認された。しかしながら、画像に回転が加わった場合の認証率は決して高いものではない。今後、回転に不変な特徴抽出法を本提案システムに組み込み、汎用性の向上を検討する必要がある。

## 6章 謝辞

1年半の間、直接ご指導頂いた竹田史章教授、プログラミングの講義をして頂いたグローリー工業(株)の西蔭様、ならびに掌紋の採取に協力して頂いた研究室の学生一同に深く感謝いたします。

## 7章 参考文献

- (1)内田薫,“バイオメトリクスによる個人認証技術とその応用”,電学誌,120 巻 7号,p.407~410,2000年
- (2)小畑秀文,“個人識別技術の現状と展望”,システム制御情報学会誌,Vol.35,No.7,pp383~389,1991
- (3)増田功,“セキュリティにおける個人識別技術”,システム制御情報学会誌,Vol.35,No.7,pp431~439,1991
- (4)“バイオメトリクスの概要”,URL=<http://www.secugen.co.jp/biometrics.html>”
- (5)“複数のバイオメトリクス認証とPKIを連携させるネットマークス”,URL=<http://www.atmarkit.co.jp/news/200005/30/netmarks.html>”
- (6)“バイオメトリック認証(Biometric Identification)”,日経BP デジタル大事典2000-2001年版,URL=<http://findx.nikkeibp.co.jp/static/yogo.html>”,(検索キーワード 指紋)
- (7)“個人認証(Personal Authentication)”,日経BP デジタル大事典2000-2001版,URL=<http://findx.nikkeibp.co.jp/static/yogo.html>”
- (8)“コンパクト指紋認証システム”,URL=[http://tech.otsuka-shokai.co.jp/a\\_secure/solution/lan/fprint/](http://tech.otsuka-shokai.co.jp/a_secure/solution/lan/fprint/)”
- (9)“バイオメトリクス元年“SecuGenのIT戦略”“,日刊工業新聞,2000年3月30日掲載
- (10)西蔭紀洋,竹田史章,吉田与志一,仁木章人“小規模ニューラルネットワーク群による筆記者認証装置”,電子情報通信学会論文誌D-,投稿中
- (11)竹田史章,西蔭紀洋,藤田靖“自己学習型ニューロ紙幣識別ボードの開発とその汎用展開”,電学論C,121巻1号,2001年
- (12)麻生英樹,“ニューラルネットワーク情報処理”,産業図書株式会社,1988年6月20日