古文書文字列に対するキャラクタスポッティング
Character Spotting for Histrical Document Character String

橋本 智広　梅田 三千雄
Tomohiro HASHIMOTO , Michio UMEDA

大阪電気通信大学大学院
〒572-8530 大阪府寝屋川市初町 18-8
Graduate school,Osaka Electro-Communication University,
18-8 Hatsu-cho,Neyagawa-shi,Osaka 572-8530,Japan

あらまし:

本論文では，古文書文字列を対象として，古文書特有のつづけ字や食い込みに対処するために，認識過程を導入した文字切り出し手法を提案する．また，文字列から任意に指定する文字のみを抽出するキャラクタスポイッティングに注目し，指定文字のみを認識して抽出する手法を提案する．まず，初期の文字切り出しとして，文字列に対するラベリング処理により，連結成分ごとに領域分割する．この領域を矩形で囲み，高さや横幅といった矩形情報を基に個々の文字パターンを切り出す．次に，切り出された文字パターンに対し，個別文字認識する．この認識結果から切り出し失敗矩形を検出し，それに対し認識処理結果に基づく再文字切り出しをする．これによって，認識結果を考慮した切り出しが可能となり，最適な位置での切り出しが期待できる．その後，抽出対象とする文字のニューラルネットワークを用いて各文字パターンの誤差を算出し，抽出条件に適合する文字パターンをスポッティング結果として選出する．本手法において，「天保郷帳」を例にとった 615 個の古文書文字列を対象とした抽出実験では，抽出対象を 5 文字として，再文字切り出しを付加することにより，導入前には 87.58%であった平均抽出率が，導入後は 94.22%に向上した．

Summary:

This paper proposes a character segmentation and spotting method of histrical documents. In the segmentation method,the result of character recognition process is utilized to cope with the cursive scripts and the mutual encroachment of characters which are peculiar to the historical documents. In the spotting method,the previously assigned characters are only extracted from the characters string. As an early segmentation,the characters string pattern is divided into the same connected component by using the labelling processing. The area composed of the same component is surrounded with a rectangle and each character pattern is segmented each other by using the shape of rectangle such as height and width. Next,the individual character recognition is applied to the segmented pattern. From the recognition result the rectangle failed in the segmentation is picked up and the resegmentation is applied to the string contains this rectangle. Therefore,it is expected that the string is divided at the best position. On the other hand the neural network which corresponds to the previously assigned character is prepared. The error between input and output of the network applied to the segmented pattern is calculated and the pattern which satisfies the condition is extracted as a spotting result. From the extraction experiment applied to 615 characters strings,the correct extraction rate of 94.22% was obtained to 5 assigned characters by using the resegmentation prosess,but the rate was 87.58% without the resegmentation process.

キーワード:

文字認識, キャラクタスポッティング, 文字切り出し, 自己想起型ニューラルネットワーク, 古文書

Keywords:

character recognition,character spotting,character segmentation,autoassociative neural network,histrical document

1 はじめに

手書き文字認識に関する研究は，様々な研究機関で試みられており，数多くの認識手法が提案され，その技術は実用の段階にある．一方，人文学研究分野では，古文書を対象としたOCRの実現を目指し，古文書に対する認識手法の提案が期待されている [1].

現在，古文書画像データベースの構築においては，史料の解読，文字データ入力に長時間の作業を必要とする．この作業を自動化できれば，飛躍的に作業時間を短縮でき，多量の古文書史料を短時間で，効率よくデータベース化できる．そこで，古文書を対象としたOCRの研究が進められている [2]-[4].

しかし，古文書を認識対象とすると，文字のパターン数に制限があるため，認識に使用する辞書作成において，十分なデータ採取が困難となる．そのため，認識対象となる文字が限定されてしまう．また，手書き文字認識のように認識に使用する辞書作成において，不足するデータを新たに作成するといったことが不可能である．従って，限られたデータの範囲内で，古文書独自の認識手法を新たに考案する必要がある．また古文書では，認識手法とともに，文字列から個々の文字パターンに切り出す文字切り出しが重要となる．しかし，古文書特有のつづけ字や文字の食い込み等から正確な文字切り出しが困難で，それに伴い高い認識結果を得にくいなどの問題もある．

本論文では，文字列を認識するだけではなく，文字列から任意に指定する文字を抽出する技法であるキャラクタスポッティングに注目し，古文書文字列から任意に指定する文字のみを抽出する手法を提案する．同時に，古文書特有のつづけ字や文字の食い込み等を考慮した，認識過程を導入した文字切り出し手法を提案する．

文字列から個々の文字パターンを切り出すために，まず初期文字切り出しとして，文字パターンにおける連結成分を囲む矩形情報を基に統合，分割を繰り返し，切り出し候補を得る．個別文字認識においては，特徴抽出に加重方向指数ヒストグラム特徴 [5] を用いた．また認識処理には，柔軟な情報処理と高い汎化能力を持ち，人間の学習過程をモデル化した，ニューラルネットワーク（以下NNと略す）を利用する．ここでは，古文書文字に対して有効とされる自己想起型NN[4] を用いた．さらに，認識処理結果から初期文字切り出しにおける切り出し失敗矩形を検出し，認識処理結果に基づく再文字切り出しをする．この認識処理を繰り返し実行することにより，最適な位置での切り出しが期待できる．そして，抽出対象となる文字に対応したNNを用意し，切り出された各文字パターンを入力して誤差を算出する．この誤差を閾値処理して抽出対象文字を選出し，対象となる文字パターンをスポットする．

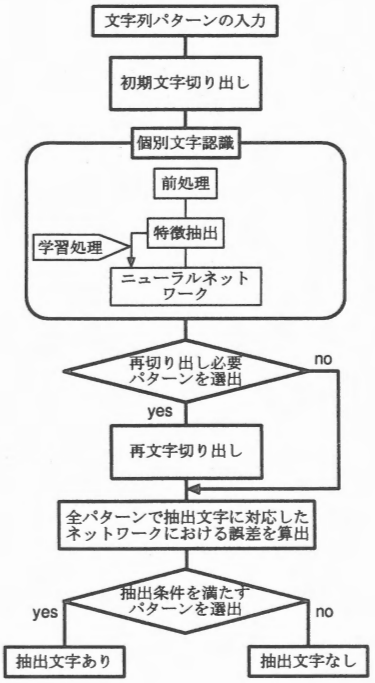

図 1: 処理の流れ

抽出実験として，「天保郷帳」を例にとり，文字列から指定する文字がどの程度抽出できるか検討するとともに，文字列からの各文字パターンに対する切り出し率と抽出率の関係について検討する．

2 システムの概要

本システムの処理手順を図1に示す．まず，対象文字列において文字を個別パターン化するために初期文字切り出しをする．これによって，切り出された各文字パターンに対し個別文字認識する．ここでは，前処理，特徴抽出を施し，さらに学習処理によってNNを形成する．初期文字切り出しでは，誤って切り出された文字パターンが存在することが多い．そこで，該当する文字パターンに対し，認識過程を導入した再文字切り出しをする．最終的に，対象文字のNNで各文字パターンにおける誤差を算出し，閾値処理によって対象文字を選出する．

3　初期文字切り出し

文字列から指定する文字のみを抽出するためには，個別に文字を切り出す必要がある．初期文字切り出しは，文字パターンにおける連結成分の高さ (*heigh*) や横幅 (*width*)，面積 (*area*) 等の情報に着目した切り出し手法である．

処理手順は，まず，ラベリングにより文字パターンの連結成分に対する外接矩形を求める．外接矩形とは，連結成分を囲む長方形のことである．この時点では,「八」や「三」のような複数の領域から構成される文字は，それぞれ連結性がないために各領域は独立している．そこで，各領域をグルーピングすることで一つの領域とするために統合処理をする．

統合方法は，4 段階に分けて行う．対象文字列が縦書きであるため，先に述べた「八」などの文字は横方向の矩形同士をグルーピングすることで一文字となる可能性が高い．図 2 に統合処理における各段階での条件を示す．第一段階として，図 2(a) のように対象矩形の高さ内に別の矩形があるとき統合する．第二段階は，対象矩形と別の矩形が図 2(b) の位置関係にあり，別の矩形の高さ比率が 40 %以上のとき統合する．この二つの条件は，矩形の高さに着目した統合方法であるが，第三段階以降は矩形同士の重なり合う面積に着目して統合する．第三段階では，図 2(c) に示すように，対象矩形が別の矩形の幅よりも小さく，重なり合っている対象矩形の面積比率が 50 %以上で統合する．さらに，第四段階として，図 2(d) に示すような別の矩形が対象矩形の横幅よりも大きく，重なりあっている別の矩形の面積比率が 50 %以上のときに統合する．この第四段階に関しては，次に説明する分割処理において，一回目の分割処理が終了した時点で適用する．これらの統合条件の他に,「三」を強制的に統合する条件も与えた．

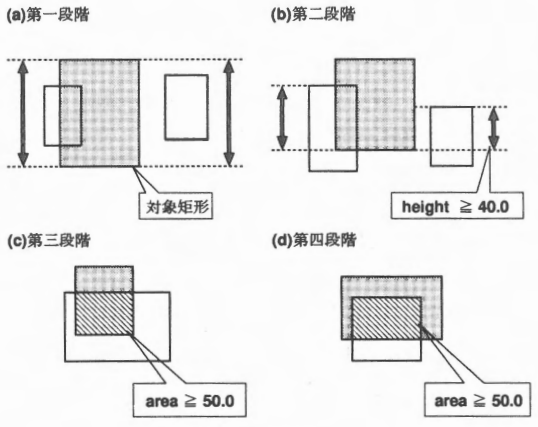

図 2: 統合方法

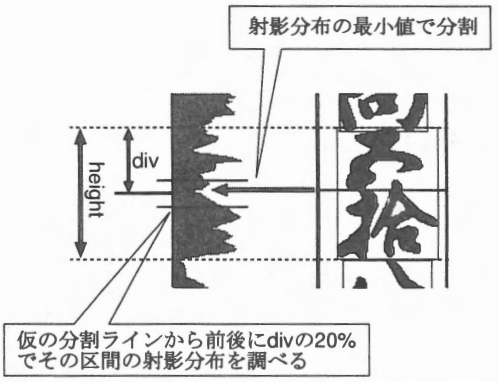

図 3: 分割方法

次に，統合処理によってグルーピングされた矩形の中には誤って統合したものや，初期の段階でつづけ字や食い込みにより，二文字以上が同一領域であるとみなされた矩形が存在する．そこで，これらの矩形に対し，分割処理をする．

分割方法は，二度の処理で完了する．まず，それぞれの分割対象矩形を選出するための条件を次のように与える．一回目の条件は，矩形内に 3 個以上の複数文字が存在すると考えられる矩形を選出するものであり，二回目は，矩形内に 2 文字程度が存在すると考えられる矩形を選出する条件である．ここで，全矩形の面積の平均を $area.ave$ とし，全矩形の横幅の平均を $width.ave$ とする．

- 分割対象矩形条件：一回目
  - $height \geq width \times 1.2$
  - $area \geq area.ave$
- 分割対象矩形条件：二回目
  - $width \geq height$
  - $width \geq width.ave$
  - $area \geq area.ave$

これらの条件にあてはまる矩形を図 3 に示すように分割する．まず初めに，$height \div width$ を求め，この値を仮の矩形内文字数 ($moji$) とする．そして，$height \div moji$ で仮の分割ライン (div) を決定する．さらに，仮の分割ラインから前後に div の 20 %の大きさをとり，その区間の射影分布を求める．そして，最小値とな

るところを最終的に分割ラインとする．二回目も同様に分割する．

4　個別文字認識

文字切り出しによって切り出された各々の文字パターンに対し，個別字認識する．まず，前処理として，孤立点除去，大きさの正規化，スムージングにより文字パターンの均一化を図る．次に，特徴抽出を行う．ここでは，数多く提案されている特徴抽出法の中から，比較的高い認識率が期待できる加重方向指数ヒストグラム特徴により特徴量を得る．さらに，得られた特徴量をもとに，NN を形成するための学習処理を行う．

4.1　前処理

切り出した個々の文字パターンは大きさにばらつきがある．そのため，各文字パターンの均一化を図るために前処理を施す．まず，画像に含まれている雑音を除去する孤立点除去，ばらつきのある大きさを均一にする大きさの正規化，さらに，大きさの正規化によって凹凸の激しくなった文字の輪郭部を平滑化するスムージングを施す．図 4 に前処理前後の文字パターンを示す．

4.2　加重方向指数ヒストグラム特徴

文字の輪郭部に着目した特徴抽出法である加重方向指数ヒストグラム特徴は，次のようにして抽出する．まず，文字パターンに対し輪郭線追跡を行いながら，輪郭部に属する各画素に対し 16 の方向指数を算出する．方向指数の算出では，図 5 に示すように，注目画素と連結している前の画素から注目画素をみた方向指数と，注目画素から後の画素をみた方向指数から注目画素の方向指数を算出する．

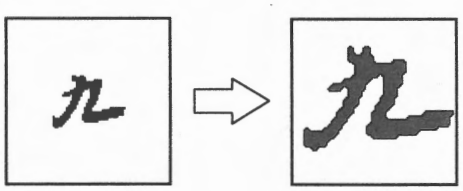

図 4: 前処理

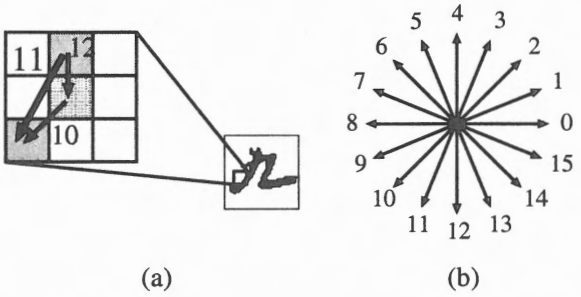

図 5: 方向指数の算出

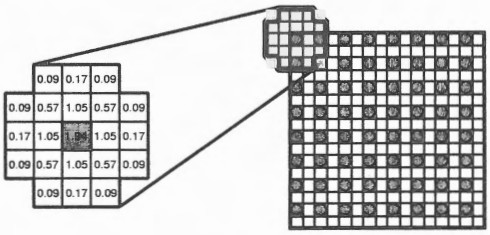

図 6: ガウスフィルタ

この例では，前の画素から注目画素を見た方向指数は 12 となり，注目画素から後の画素をみた方向指数は 10 となる．そこで，両者の方向指数の平均をとることで注目画素の方向指数 11 と算出する．

そして，方向指数を算出後，各方向指数に対して方向圧縮する．まず，奇数方向に対し重み付けし，それに対して前後の偶数方向を足しこむ処理により，16 方向から 8 方向へと圧縮する．さらに，反対方向を同一視することにより，4 方向へと圧縮する．

次に，領域圧縮として，一定領域でヒストグラムを求める．さらに，図 6 に示すような，局所的な位置をぼかす働きをもつガウスフィルタを用いて領域圧縮する．ガウスフィルタは画素一つおきにフィルタリングする．そして，ここでは 8×8 領域 ×4 方向から成る 256 次元の特徴量を得る．

4.3　自己想起型ニューラルネットワーク

認識には，その柔軟で，かつ高い汎化能力から，文字認識において利用されることが多い NN を使用する．ここでは，特に NN の中でも古文書に対して有効とされる自己想起型 NN を用いることとした．

自己想起型 NN は入力層と出力層のユニット数が等しく，入力パターンそのものを理想出力とするネッ

トワークである．従って，教師信号には入力パターンそのものを与える．ここで，NN 形成における学習とは，実際の出力と望ましい出力との差，つまり誤差を小さくするように重みを変更していくことを意味する．学習には，バックプロパゲーション法（BP 法）を用いた．

これは，出力層のニューロン値 O_k と理想的な出力である教師信号 T_k との二乗誤差を式 (1) により求め，その誤差が最小となるように出力層と中間層，中間層と入力層間のシナプスの重みを変更するものである．

$$e = \sum_k (T_k - O_k)^2 \tag{1}$$

図 7 にニューロンモデルを示す．ここで，各ユニットに対する入力の総和を

$$net_i = \sum_j W_{ij} O_j \tag{2}$$

とする．次に，そのユニットの出力 O_i をとすると，

$$O_i = f(net_i) \tag{3}$$

となる．この出力関数には式 (4) のシグモイド関数を用いる．

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

重みの変化量は学習係数 η とすると，

$$\triangle W = \eta \left(- \frac{\partial E_p}{\partial W} \right) \tag{5}$$

で求めることができる．ここで，出力層と中間層の重みの変化量は，

$$- \frac{\partial E_p}{\partial W} = (T_k - O_k) f'(net_k) O_j \tag{6}$$

$$\delta_k = (T_k - O_k) f'(net_k) \tag{7}$$

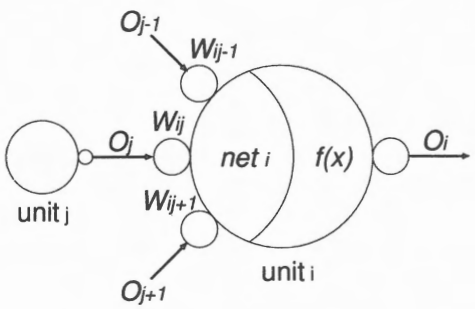

図 7: ニューロンモデル

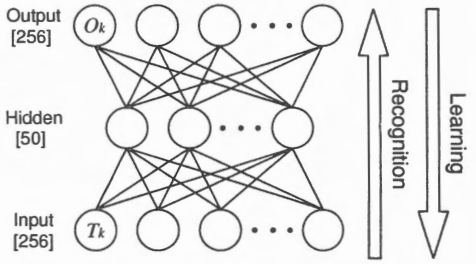

図 8: 自己想起型ニューラルネットワーク

より，

$$\triangle W_{kj} = \eta \delta_k O_k \tag{8}$$

で与えられる．また，中間層と入力層との間では，

$$- \frac{\partial E_p}{\partial W} = f'(net_j) \sum_k (\delta_k W_{kj}) O_i \tag{9}$$

$$\delta_j = f'(net_j) \sum_k (\delta_k W_{kj}) \tag{10}$$

とすることにより変化量は，

$$\triangle W_{ij} = \eta \delta_j O_i \tag{11}$$

となる．

このような重みの変化量を用いて，出力層と教師信号の間の誤差が小さくなるように，各ユニット間の重みを変更していくことで NN を形成する．

図 8 に，ここで使用したネットワーク構成を示す．各層のユニット数は，入力層と出力層が 256 個，中間層は 50 個とした．また，このネットワークはカテゴリごとに形成するため，カテゴリの変化に容易に対応できる特徴がある．そのため，同時に他の文字の影響を受けない学習が可能となる．

5 認識過程を導入した再文字切り出し

初期文字切り出しでは，文字パターンの連結成分における矩形情報のみを利用するため，完全に文字が個々のパターンに切り出されていないことが多くある．ここで提案する切り出し手法は，古文書特有のつづけ字や文字の食い込みなどを考慮し，認識処理結果に基づき再文字切り出しすることによって，高精度な切り出しを実現する．

本手法では，次に示す条件を基に再切り出し候補矩形を求める．

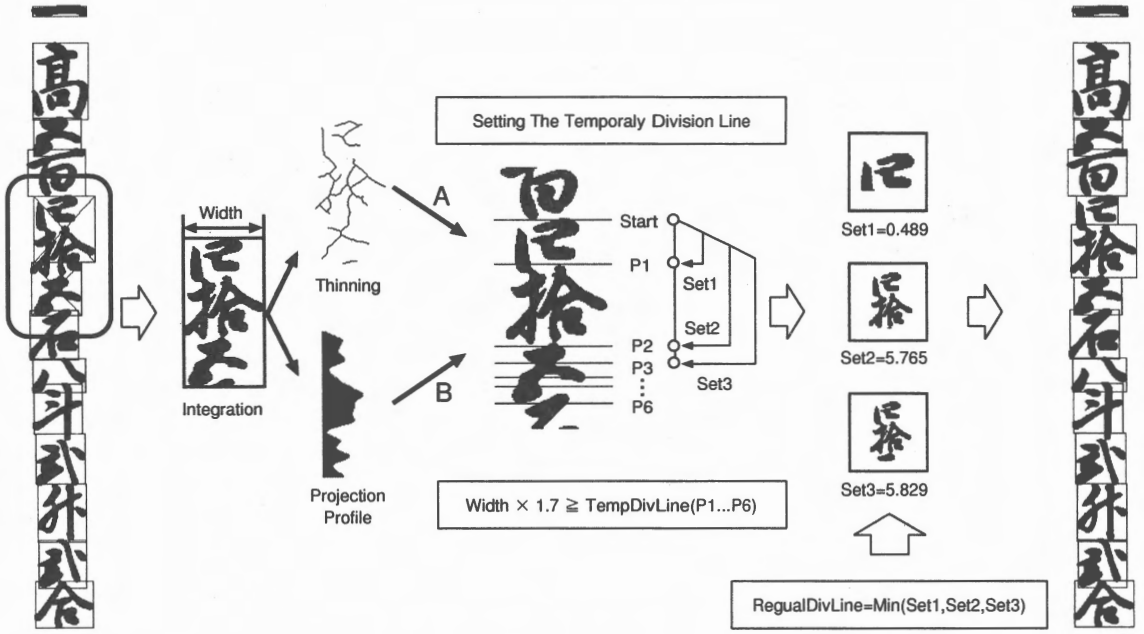

図 10: 分割ラインの決定方法

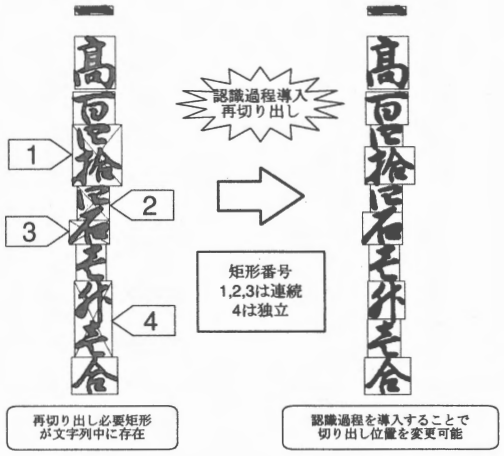

図 9: 再切り出し必要矩形

- 条件 α : NN における二乗誤差 ≥ 2.5
- 条件 β : NN における二乗誤差 ≥ 1.5

まず，切り出した各矩形に対して個別文字認識する．このとき，NN における誤差をもとに，条件 α を満たす矩形は誤って切り出されたとして再切り出し必要な矩形とする．ここで，全矩形において再切り出しが必要とされる矩形の位置関係は，次のことが挙げられる．

- 再切り出しが必要とされた矩形の上下矩形のどちらも再切り出し不要とされる場合（再切り出し必要矩形が独立）
- 再切り出しが必要とされた矩形の上下矩形のいずれかが再切り出し必要とされる場合（再切り出し必要矩形が連続）

図 9 に再切り出し必要とされた矩形例を示す．ここでは，初期文字切り出しの段階で得られた切り出し候補に対し，条件 α を満たしていない矩形 4 個に目印となる ”×” が印されている．それぞれを矩形番号 1,2,3,4 とすると，1,2,3 においては連続していることがわかる．一方，4 は上下の矩形には目印が付いていないため独立している．そこで，再切り出し必要矩形に対し再文字切り出しを適用する．

ここで，連続している矩形の位置関係から互いに少しでも離れている場合は独立しているとする．それ以外のときは，強制的に連続している矩形を統合する．図 9 では，1,2,3 の矩形の位置関係は離れていないために全て統合する．そして，統合後の矩形に対して個別文字認識する．このとき，条件 α を満たさなければ統合することによって適切に切り出された矩形とする．しかし，条件 α を満たす矩形に対しては再分割する．これ以降の処理は再切り出し必要矩形が独立している場合も同様にする．

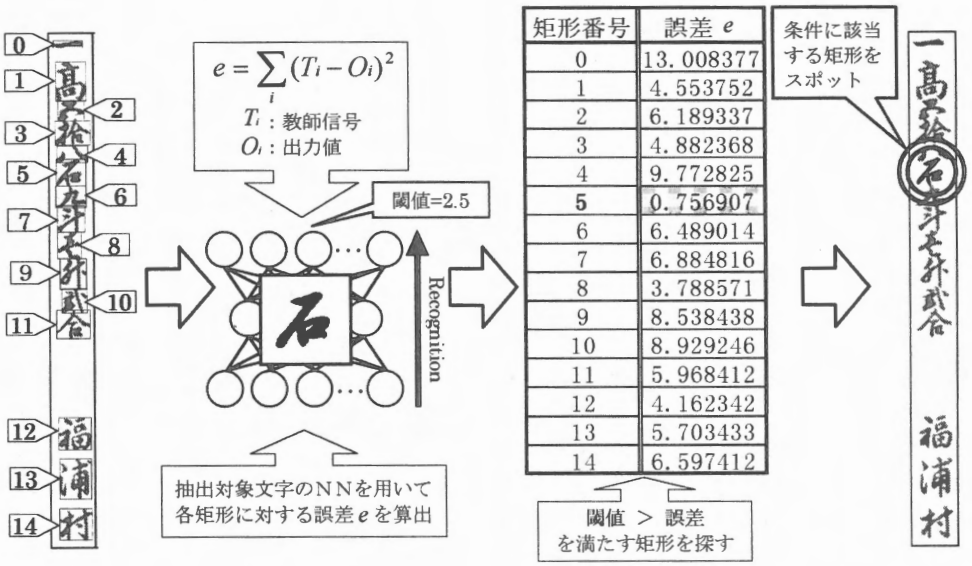

図 11: スポッティング処理

図 10 に再分割処理手順を示す．まず，対象となる矩形に対し，Hilditch の細線化処理 [6] によって矩形内の文字パターンを線幅 1 で表し，横方向の射影分布を求める．このとき，射影値 1 となるところは文字同士が食い込んでいないところである．そこで，射影値 1 となる部分をピックアップし，これを仮の分割ライン A とする．しかし，これだけでは文字の食い込みに対応できない．そこで，原画像における射影分布を求め，分布の平滑化を図るとともに，射影値を一定の割合で小さくする．

ここで，あらかじめ対象矩形において，収縮法 [7] により文字の線幅を求めておく．求めた線幅を閾値として，射影値が閾値より小さい部分を仮の分割ライン B とする．以上の処理により仮の分割ラインを決定する．しかし，これでは，仮の分割ラインが対象とする矩形によっては複数になることが考えられる．そこで，仮の分割ラインが連続している場合はその中間点を最終的な仮の分割ラインとする．また独立している場合はそのままとする．これにより仮の分割ラインを最小限に抑えることができる．

仮の分割ラインから分割ラインを決定するとき，矩形の最上部から各々の仮の分割ラインまでの領域に対し個別文字認識する．その際，認識における誤差が最小となるところを分割ラインとする．

まず，文字列中の一文字あたりのサイズには限界があるため，対象矩形の高さに着目して，最上部から矩形幅の 1.7 倍以内にある仮の分割ラインについて誤差をそれぞれで求める．図 10 では，仮の分割ラインが 7 本ある．そこで矩形幅 *width* の 1.7 倍以内にある分割ラインを調べると，Start から P3 までが該当する．そのため，Start を基準として P1,P2,P3 の分割ラインまでの領域をそれぞれ Set1,Set2,Set3 として認識する．そして，認識における誤差が最小となるところを分割ラインとするため，Start から P1 までの領域が一つの文字パターンであるといえる．それ以降は基準を分割ラインとした地点，すなわち P1 を Start として以後同様の処理で分割ラインを決定する．

最後に全矩形に対して個別文字認識し，条件 β に該当する矩形を見つける．そして，上下の矩形で同様に該当しているものがあれば仮統合し，認識する．その際，条件 α を満たさないならば統合する．そして，最終切り出し候補を得る．

6　文字列からの指定文字抽出

文字列から切り出しによって得られた各文字パターンにおいて，指定する文字を選出するためには，対象文字に対応したネットワークを用意する必要がある．これを利用して各文字パターンにおける二乗誤差を算出し，この誤差から対象文字を選出する．抽出文字の選出には閾値処理を適用する．これにより，

文字列中に対象文字が複数ある場合でも全て抽出することが可能となる．しかし，別の文字を抽出することも考えられるため，各文字に対する適切な閾値を決定する必要がある．

図 11 に抽出までの一連の処理の流れを示す．まず初めに，入力文字列に対し文字切り出しで得られた各文字パターンに矩形番号を付ける．そして，特徴抽出で得られた特徴量を基に抽出対象文字の NN を用意し，これを用いて各文字パターンに対する誤差 e を算出する．ここで，対象文字ごとに閾値を設定し，条件 **閾値＞誤差** を満たす矩形を探す．最終的に，該当矩形の文字パターンを対象文字として選出する．

仮に，文字列より「石」を抽出しようとする．まず,「石」に対する NN を用いて各文字パターンにおける誤差を算出する．このとき，閾値を 2.5 と設定しているため，図 11 中の表から誤差が 2.5 より小さな矩形を探すと，矩形番号 5 が条件を満たす．そして，この矩形を対象文字として選出する．

7　抽出実験

実験用データとして，江戸時代に書かれた書物「天保郷帳」に収められている相模国に該当する，当時の各村における石高を示した文字列 615 個を用いた．これらはイメージスキャナで解像度 500dpi で採取し，一文字列あたりのサイズは 1140 × 100pixel である．図 12 に文字列例を示す．各文字列は前半に石高，続いて該当する村が記されている．また，対象文字を文字列から任意に 100 パターン選出し，ニューラルネットワーク形成のための学習パターンとして使用する．ここでは，文字切り出しの段階で比較的文字パターンが正確に外接矩形で囲まれている（切り出されている）ものを選出した．なお，学習回数は 200 回とした．

実験は，文字列に含まれる文字のうち，石高表記に用いられる「石, 斗, 升, 合」と「村」の 5 文字を対象文字とした．そこで，対象文字のみを正確に抽出するためには，各々の対象文字において最適な誤差の閾値を決定する必要がある．ここでは，閾値を 6 段階に変化させて抽出率の変化を調べた．また，閾値の設定に伴い，正確に対象文字のみを抽出する場合や，別の文字を誤って抽出する場合などが考えられる．そのため，閾値の変化により文字列から指定する文字がどの程度抽出可能かについて実験した．

各対象文字における抽出実験結果を表 2 から表 6 に示す．なお，各表の項目内容（内訳）を表 1 に示しておく．ここで，抽出率，切り出し率は

$$\text{抽出率} = \frac{\text{対象文字のみ抽出した数}}{\text{対象文字を含む文字列総数}} \times 100$$

$$\text{切り出し率} = \frac{\text{対象文字が切り出されている数}}{\text{対象文字を含む文字列総数}} \times 100$$

と定義した．抽出率とは，対象文字を含む全文字列において，対象文字のみを抽出したときの文字列数の割合であり，切り出し率とは，対象文字を含む全文字列において，文字切り出しの段階で，指定する文字パターンが正確に外接矩形で囲まれている文字列数の割合とした．さらに，各対象文字の閾値に対する抽出率の変化を図 13 に示す．このときの各対象文字において，最高の抽出率を得るときの閾値がその文字の最適な閾値となる．

各文字の抽出率の変化をみると，閾値が小さい段階では抽出率も低いが，閾値を大きくするにつれて抽出率は高くなる．これは，対象文字が適切に切り出せていない矩形に対し，抽出条件が満たされていないためである．また，正確に切り出せている矩形に対しては低い段階でも正確に抽出できる．一方，閾値を大きくすることで，あるところから抽出率は低下する．これは，対象文字が抽出されるのと同時に，別の文字も抽出するためである．

さらに，切り出し率と抽出率の関係を調べるために，各対象文字での切り出し率と抽出率を表 7 に示す．また，それぞれ最適な閾値もあわせて示す．これより，切り出し率と抽出率は大きく関係するといえる．その中でも，切り出しの段階で高い切り出し率を得ている「村」に関しては，抽出率も同様に高くなることがわかる．また，石高部において後に続く

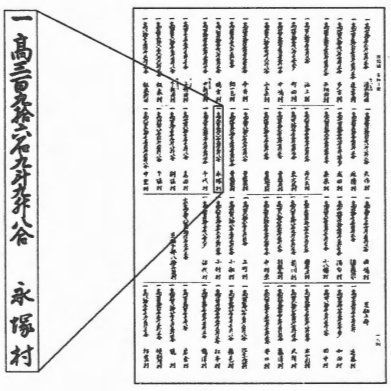

図 12: 使用データ

表 1: 結果表の内訳

内訳	A	対象のみスポットした
	B	対象と別の文字をスポットした
	C	別の文字のみスポットした
	D	どの文字もスポットしなかった

表 2: 「石」に対する抽出結果

閾値	A	B	C	D	抽出率 (%)
1.00	464	0	0	151	75.45
1.50	540	0	0	75	87.80
2.00	561	0	0	54	91.22
2.50	572	2	0	41	93.01
2.75	572	6	0	37	93.01
3.00	563	16	2	34	91.54

表 3: 「斗」に対する抽出結果

閾値	A	B	C	D	抽出率 (%)
1.00	448	0	0	100	81.75
1.50	497	0	0	51	90.69
2.00	516	5	0	27	94.16
2.13	519	7	1	21	94.71
2.25	515	16	2	15	93.98
2.50	491	47	0	10	89.60

表 4: 「升」に対する抽出結果

閾値	A	B	C	D	抽出率 (%)
1.00	341	0	0	188	64.46
1.50	436	0	0	93	82.42
2.00	468	0	0	61	88.47
2.50	482	9	0	38	91.12
2.63	475	18	3	33	89.79
2.75	463	29	3	34	87.52

表 5: 「合」に対する抽出結果

閾値	A	B	C	D	抽出率 (%)
1.00	346	0	0	189	64.67
1.50	453	0	0	82	84.67
2.00	489	0	0	46	91.40
2.50	500	8	0	27	93.46
2.75	495	13	1	26	92.52
3.00	486	24	1	24	90.84

表 6: 「村」に対する抽出結果

閾値	A	B	C	D	抽出率 (%)
1.0	560	0	0	31	94.75
1.5	580	0	0	11	98.14
2.0	584	0	0	7	98.82
2.25	581	4	0	6	98.31
2.38	573	12	0	6	96.95
2.5	564	21	0	6	95.43

表 7: 総合評価結果

対象文字	切り出し率 (%)	抽出率 (%)	閾値
石	89.92	93.01	2.50
斗	88.69	94.71	2.13
升	82.99	91.12	2.50
合	93.64	93.46	2.50
村	97.80	98.82	2.00

文字が少ない「合」は，他の文字と比べると，切り出し率は高くなっている．一方,「升」は切り出し率，抽出率ともに低くなった．これは,「升」は前後の文字と重なっている場合が多いことや，文字パターンの射影分布が平均していることで，切り出し位置の推定に失敗したことが切り出し率の低下につながったといえる．

切り出しに失敗した文字列例を図 14 に示す．該当する文字列は，対象文字の外接矩形内に別の文字が存在しているため，正しく囲まれていないことがわかる．このような文字パターンは切り出し失敗とした．

次に，再文字切り出しの効果を確かめるために，導

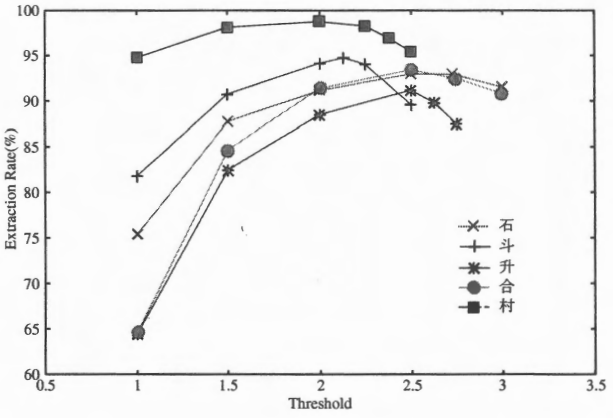

図 13: 各対象文字の閾値に対する抽出率の変化

表 8: 切り出し過程の違いによる抽出率の変化

対象文字	認識導入前 (%)	認識導入後 (%)
石	84.23	93.01
斗	87.59	94.71
升	80.72	91.12
合	86.54	93.46
村	98.82	98.82

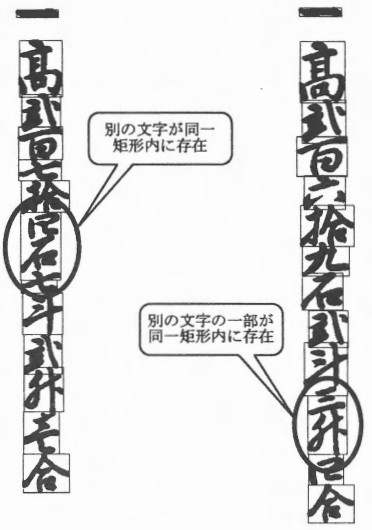

図 14: 切り出し失敗文字列

入前と導入後の各対象文字の切り出し率と抽出率を比較する．表 8 にその結果を示す．これより，文字列において，再文字切り出しを適用する部分は石高部のみであるため,「村」に関しては変化がみられない．しかし，その他の文字では導入前と比べると，導入後の抽出率共は高い値を示すことが見受けられる．これは，切り出しの段階で認識過程を導入したことの効果であるといえよう．

8 おわりに

本論文では，つづけ字や食い込み等が原因で，前後の文字が互いに影響しあう古文書文字列に対し，これらを考慮した文字切り出し手法を提案し，任意に指定した文字のみをいかに文字列から抽出できるかについて検討した．

その結果，提案手法では抽出対象文字の平均抽出率は 94.22%，平均切り出し率は 90.66%が得られた．これより，文字切り出しの段階で認識過程を導入したことが，切り出し率の向上つながったことがわかる．また，これに伴って高い抽出率が得られたといえよう．

キャラクタスポッティングでは，個々の文字を正確に認識するだけではなく，いかに適切に文字列から個々の文字を切り出すかという前段階が重要であるとわかった．今後は，毛筆書体の文字列認識を進めるにあたって，キー文字の抽出技術を向上させ，認識の高精度化につなげる検討を加える必要がある．また，他の書体の古文書にも応用できるように検討しなければならない．

参考文献

[1] 山田奨治 ”古文書 OCR 研究の現在 ” 挑戦古文書 OCR，人文学と情報処理，No.18，pp.2-5，1998.

[2] 日置慎治，上原邦彦，川口洋 ”「宗門改帳」に記録された年齢表記の認識 ” 挑戦古文書 OCR，人文学と情報処理，No.18，pp.35-42，1998.

[3] 和泉勇治，加藤寧，根元義章，山田奨治，柴山守，川口洋 ” ニューラルネットワークを用いた古文書個別文字認識に関する一検討 ” 情報処理学会研究報告，Vol.2000，No.8，pp.9-15，2000.

[4] 橋本智広，横田宏，梅田三千雄 ”自己想起型ニューラルネットワークによる古文書文字認識 ” 電気関係学会関西連合大会論文誌，G13-14，2000.

[5] 鶴岡信治，栗田正徳，栗田昌徳，原田智夫，木村文隆，三宅康二 ” 加重方向指数ヒストグラム法による手書き漢字・ひらがな認識 ” 電子情報通信学会論文誌，Vol.J70-D-Ⅱ，No.7，pp.1390-1397，1987.

[6] 手塚慶一，北橋忠宏，小川秀夫 ”ディジタル画像処理工学 ”，pp.139-142，日刊工業新聞社，1985.

[7] 加藤寧，根元義章 ” ストローク情報に基づく手書郵便宛先の切り出しと認識 ” 画像ラボ，Vol.8，No.8，pp42-45，1997.