

パターン認識用プログラムについての研究

八 木 寛
高 田 正 一 郎
三 木 修 武

On a Research for Pattern Recognition Program

Hiroshi YAGI
Shoichiro TAKATA
Osamu MIKI

This paper reports a kind of pattern recognition. There are many patterns in the world, such as the patterns of characters, pictures and voices.

We selected Katakana characters for our objects and studied how Katakana characters are recognized by computers. We studied two cases.

One was that the input characters were stylized and the other the input characters were unstylized. The result was that the computer could recognize all stylized characters.

1. ま え が き

近年、種々な業務において、能率化の一端として手書き文字の機械的処理が望まれている。たとえば、郵便物の自動宛先分類、銀行窓口での筆跡鑑定、電子計算機の事務処理上でのパンチングの省略等が、その代表的な例であろう。パターン認識は、その手書き文字の機械的処理化のための基本的な研究課題であろう。筆者らは、パターン認識の基礎的な知識を得るためにデジタル計算機を使って、パターンの一つとして、手書きカタカナ文字をとりあげ、それをいかに認識させるか、また、そのプログラムについて調べた。

なお、デジタル計算機はOKITAC-5090を使用し、プログラムはOKIPALによって構成されている。

2. パターン認識の過程

パターン認識とは、人が視覚、聴覚、触覚を通じて行なう図形、文字、音声の判別の働きをいう。われわれは日常、これらの事をいともたやすく行なっているが、その内容は、非常に複雑な情報処理であって、その過程は、現在解明されるに至っていない。しかしながら、パターン認識は、大体、図-1のような認識過程をたどっていると考えられる。

すなわち、その過程は、入力パターンの観測過程、特徴抽出過程、識別過程の3者に大別される。筆者らは、3者のうち入力パターンの観測過程を除く特徴抽出過程、識別過程を研究の対象とした。パターンを認識する機械は、一般的に3者で構成されるとし、3者の機能は次のようになる。

2・1. 入力パターンの観測過程

未知パターン（入力パターン）は機械的方法、電気的方法、両者の組み合わせによる方法のいずれかの方法によって、装置の入力部に移し、未知パターンが、光学的、電氣的に観測され、2・2で記す特徴パラメータの抽出が行なわれ易いように正規化（normalize）される。正規化とは観測された未知パターンの必ずしも統一されない位置、大きさ、傾き、太

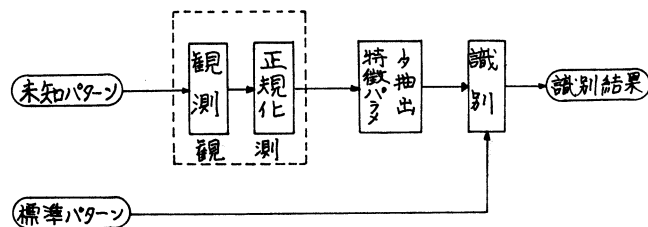


図-1 パターン認識過程

さ、濃さを一律化する過程である。

2・2. 特徴抽出過程

観測過程で正規化された入力パターンより特徴パラメータを抽出する。特徴パラメータとは、文字の場合についていえば、縦線、横線、斜め線の有無、または本数、上部への突出部の有無、または本数等の特徴をスコア化したものである。この特徴パラメータの質と量は、次の識別過程において、誤認識、認識可能、不可能を左右する重要な要素である。なお、数量に関しては、対象とするパターンを類別するに必要かつ最小限でなければならない。特徴パラメータの数量が多すぎたり、その質が厳密すぎると、変形した入力パターンに対しては、適合する回数が少なくなって、認識不能、あるいは拒否 (reject) の機会が多くなる。また、逆に特徴パラメータの数量が少なすぎると誤認識の機会が多くなるのである。

2・3. 識別過程

記憶装置にあらかじめ、標準パターンの特徴パラメータを記憶させておき、それと特徴抽出過程で抽出した未知パターンの特徴パラメータと比較する。なお、標準パターンの特徴パラメータと未知パターンのその数は等しくなければならない。比較されて両者が等価であるとき、標準パターン、あるいはそのシンボルを識別結果として、打ちださすのである。

筆者らは、以上のような過程を有するよう、パターン認識用プログラムを作り、それを特徴抽出法と呼んだ。

3. 特徴抽出法

3・1. 対象とする文字パターン

入力パターンの対象をカタカナ48文字（濁音は除かれている）とした。入力パターンは図-2のように10×10のマトリックスの形で文字の成分がある部分は“1”，空白は“0”と表わされている。

3・2. 特徴の種類

標準特徴パラメータの質と数量は、2・2で記したように、定めねばならない。筆者らは、カタカナ文字を観察して、次のように特徴を選出し分類した。

(1) 水平成分の本数。

(2) 垂直成分の本数。

(3) 斜め成分 (＼) の有無。(＼は右下りの斜線)

(4) 斜め成分 (／) の有無。

(5) 文字の重心が上部，下部のいずれの部分にあるか。

(6) 文字の重心が左部，右部のいずれの部分にあるか。

(7) 上部に突出部があるか，またその本数。

(例えば“サ”であれば突出部は2本)

(8) 中心部に空部があるかどうか。

(例えば“タ”であればなし，“ク”であればある)

(9) “ヌ”と“ズ”のように突出しているか否か。

以上のような特徴があげられるが、(1)～(6)の特徴は

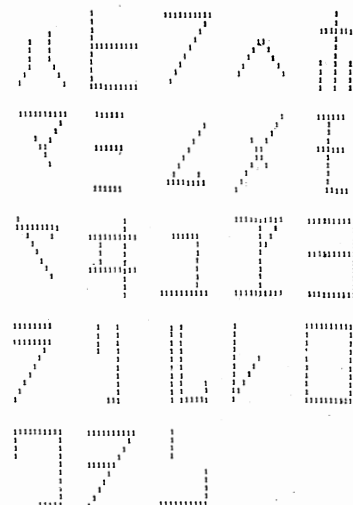
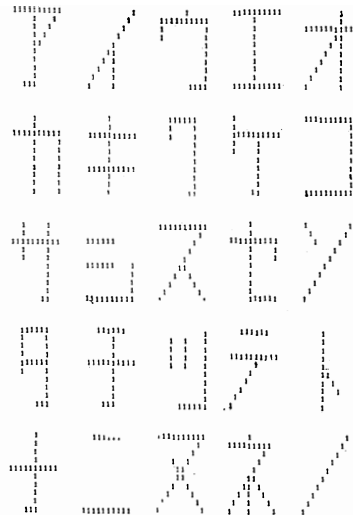


図-2

図-3 標準パターン

48個のカタカナ文字に対して、普遍的であり、識別に際して充分安全性を持つ。

(7)~(9)の特徴は必ずしもそうではない。

標準特徴パラメータのリストを作ろうとする際、まず標準パターン48個のカタカナ文字(図-3)を(1)~(6)について調べ、その結果をスコア化する。それで、各々の文字は、6個の特徴のスコアの組み合わせでもって代表される。(例えば“セ”であれば(2 2 0 0 1 1)のように)しかし、6個のスコアの組み合わせでは、48文字は分類できず、一つの組み合わせに対し2個以上の文字の対応が生じる。が、その場合に限りそれを例外(Exception)として、(7)~(9)のいずれかの特徴について調べ、その結果をスコア化する。結局高々、8個の特徴パラメータの組み合わせでもって、48個の文字に対応する組み合わせは、各々、異なるようになった。その特徴パラメータのリストは表-1である。

表-1

ア	1 1 0 1 1 1 0 0	ハ	1 2 1 1 2 2 0 0
イ	0 1 0 1 2 2 0 0	ヒ	2 1 0 0 0 1 1 1
ウ	1 1 0 0 1 2 1 0	フ	1 0 0 1 1 1 0 0
エ	2 1 0 1 1 2 0 0	ヘ	0 0 1 1 2 0 0 0
オ	1 1 0 1 1 2 0 0	ホ	1 3 0 0 2 2 0 0
カ	1 2 0 0 1 1 0 0	マ	1 0 1 1 1 2 1 0
キ	2 1 0 0 0 1 1 0	ミ	3 0 0 0 1 0 0 0
ク	1 1 0 0 1 2 0 0	ム	1 0 1 1 2 2 0 0
ケ	1 2 0 0 1 2 1 0	メ	0 0 1 1 0 0 0 0
コ	2 1 0 0 0 2 0 0	モ	3 1 0 0 1 1 0 0
サ	1 2 0 0 1 2 2 0	ヤ	1 0 1 0 1 1 0 0
シ	3 1 0 0 2 1 0 0	ユ	2 1 0 0 2 2 0 0
ス	1 0 1 1 1 2 0 1	エ	2 1 0 1 1 2 0 0
セ	2 2 0 0 1 1 0 0	ヨ	3 1 0 0 1 2 0 0
ソ	0 0 0 1 1 1 0 0	ラ	2 0 0 1 1 1 0 0
タ	2 2 0 0 1 2 1 0	リ	0 2 0 0 1 2 0 0
チ	2 1 0 0 1 2 0 0	ル	1 2 0 0 2 1 0 0
ツ	1 3 0 0 1 2 0 0	レ	0 1 0 1 2 1 0 0
テ	2 0 0 1 1 1 0 1	ロ	2 2 0 0 0 0 0 0
ト	0 1 1 0 2 1 0 0	ワ	2 2 0 0 1 2 0 0
ナ	1 1 0 0 1 2 1 1	ヲ	2 0 0 1 1 1 1 1
ニ	2 0 0 0 2 0 0 0	ン	1 2 0 0 2 0 0 0
ヌ	1 0 1 1 1 2 1 1		
ネ	1 1 1 1 2 1 0 0		
ノ	0 0 0 1 0 0 0 0		

なお、特徴パラメータのスコア化の規則は次のようである。(1)、(2):本数をスコアとする。(3)、(4):有を1、無を0とする。(5):上部にあれば1、下部にあれば2、中間部にあれば0とする。(6):左部にあれば1

右部にあれば2、中間部にあれば0とする。(7):本数をスコアとする。(8):空部があれば0、なければ1とする。(9):突出していれば1、突出していなければ0とする。

3・3. 特徴抽出用プログラムについて(フローチャートプログラムを参照)

3・2で記した例外(Exception)は6個生じた。この6個の例外*(プログラムでは $Z_1, Z_2 \dots Z_6[1:6]$)をあらかじめ、記憶させておく。また、標準パターンのデータと3・2の規則に従って定めた標準特徴パラメータ(表1)のデータとを讀ませ、磁気テープに記憶させておく。プログラムでは、両者はAA[1:12, 1:10]で表わされ、AA[1:10, 1:10]は標準パターンのデータでありAA[11, 1:8]が標準特徴パラメータであり、残るは0のスコアとなっている。

次に未知パターン(XX[1:10, 1:10])を讀み、3・2の(1)~(6)の特徴について調べ、3・2の規則に従ってスコアを与える。できた6個のスコア(CH[1], CH[2]……CH[6])を、あらかじめ讀んだ6個の例外のいずれかに相当しないか調べる。いずれかの例外に相当した場合は、3・2の(7)~(9)のいずれか1個、または2個の特徴について調べ、やはりスコアを与える。(CH[7], CH[8])。また、いずれにも相当しなかった場合は、CH[7]=CH[8]=0とスコアを0とする。

結局、未知パターンの特徴パラメータ、CH[1], CH[2]……CH[8]が計算されたことになる。次に磁気テープに納まる標準パターンとその標準特徴パラメータのデータ(AA[1:12, 1:10])を一文字ずつ読みだし、その度、標準特徴パラメータ(AA[11, 1], AA[11, 2]……AA[11, 8])と計算された未知パターンの特徴パラメータ(CH[1], CH[2]……CH[8])とを比較する。*両者が等しい時、標準パラメータに相当する標準パターンを印刷させるのである。未知パターンの特徴パラメータに相当する標準特徴パラメータがなかったとき、その未知パターンは認識されなかった事とし、“NINSHIKI, FUKANO”とうたせている。

次の未知パターンに対しては、以上のことを繰返すのである。

次にプログラムおよびフローチャートをかかげる。

* (110012):ウクナ (210001):エキヒ (120012):ケサ (101112):スヌマ (220012):タク (200111):テラヲ

* $SUM = |CH[1] - AA[11, 1]| + |CH[2] - AA[11, 2]| + \dots + |CH[8] - AA[11, 8]|$ を計算し、 $SUM = 0$ であれば未知パターンはその標準パターンと等しく、 $S \neq 0$ であれば両者はそうでない。

また、認識結果の1例をあとにかかげる。

プログラム

```

begin integer A1, MTADDR, I, J, S, S1, S2, SG, K, W, P, E, SUM;
  integer array Z1, Z2, Z3, Z4, Z5, Z6, ZX[1:6], AA[1:12, 1:10], XX[1:10, 1:10], CH[1:8],
  AR[1:10] ;
  Read array (Z1[1:6]) ; Read array (Z2[1:6]) ; Read array (Z3[1:6]) ; Read array (Z4[1:6])
  Read array (Z5[1:6]) ; Read array (Z6[1:6]) ; Read1(A1) ;
  MTADDR : = A1 ;   AA[1,1] : = 0 ;
REPTR :
  Read array (AA[1:12, 1:10]) ;
  Write tape (AA[1:12,1:10], 72, MTADDR) ; if AA[1,1]= 5 then go to REDATA ; go to REPTR;
REDATA :
  Read array (XX[1:10,1:10]) ;   if XX[1,1]= 3 then go to END3 ;
EXTCHA :
  CHARA1 : CH[1]:=0 ; for I:=1 step 1 until 10 do
    begin S:=0 ; for J:=1 step 1 until 10 do S:=S+XX[I,J] ;
      if S $\geq$ 4 then AR[I]:=1 else AR[I]:=0; CH[1]:=CH[1]+AR[I]
    end ;
  CHARA2 : CH[2]:=0 ; for J:=1 step 1 until 10 do
    begin S:=0 ; for I:=1 step 1 until 10 do S:=S+XX[I,J] ;
      if S $\geq$ 4 then AR[J]:=1 else AR[J]:=0 ; CH[2]:=CH[2]+AR[J]
    end ;
  CHARA3 : S:=0 ; for I:=1 step 1 until 10 do S:=S+XX[I,I] ;
    if S $\geq$ 4 then CH[3]:=1 else CH[3]:=0 ;
  CHARA4 :
    S:=XX[1,10]+XX[2,9]+XX[3,8]+XX[4,7]+XX[5,6]+XX[6,5]+XX[7,4]+ XX[8,3]+XX[9,2]
    +XX[10,1] ;
    if S $\geq$ 4 then CH[4]:=1 else CH[4]:=0 ;
  CHARA5 :
    S1:=0 ; for I:=1 step 1 until 5 do for J:=1 step 1 until 10 do
      S1:=S1+XX[I,J] ;
    S2:=0 ; for I:=6 step 1 until 10 do for J:=1 step 1 until 10 do
      S2:=S2+XX[I,J] ; SG:=S1-S2 ;
    if SG>0 then CH[5]:=1 else begin if SG=0 then CH[5]:=0 else CH[5]:=2 end ;
  CHARA6 :
    S1:=0 ; for J:=1 step 1 until 5 do for I:=1 step 1 until 10 do
      S1:=S1+XX[I,J] ;
    S2:=0 ; for J:=6 step 1 until 10 do for I:=1 step 1 until 10 do
      S2:=S2+XX[I,J] ; SG:=S1-S2 ;
    if SG>0 then CH[6]:=1 else begin if SG=0 then CH[6]:=0 else CH[6]:=2 end ;
EXCEPT :
  switch SW1:=EXCEP1, EXCEP2, EXCEP3, EXCEP4, END1 ;
    K:=1 ,
  EXCEP1 : for P:=1 step 1 until 6 do ZX[P]:=Z1[P] , go to KEISA1 ;
  EXCEP2 : if W=0 then

```

```

begin S:=XX[5,5]+XX[5,6]+XX[6,5]+XX[6,6]; if S=0 then CH[8]:=0 else
CH[8]:=1;
go to END2 ;
end ; for P:=1 step 1 until 6 do ZX[P]:=Z2[P]; go to KEISA1 ;
EXCEP3 : if W=0 then
begin S:=0 ; for J:=1 step 1 until 2 do for I:=1 step 1 until 10 do S:=S+XX[I,J];
if S $\geq$ 10 then CH[8]:=1 else CH[8]:=0 ; go to END2
end ; for P:=1 step 1 until 6 do ZX[P]:=Z3[P] ; go to KEISA1 ;
EXCEP4 : if W=0 then begin CH[8]:=0 ; go to END2 end ;
KEISA1 : W:=0 ; for P:=1 step 1 until 6 do W:=W+abs(CH[P]-ZX[P]) ;
if W=0 then
begin E:=0 ; I:=1; for J:=1 step 1 until 10 do E:=E+XX[I,J] ;
if E=1 then CH[7]:=1 else begin if E=2 then CH[7]:=2 else CH[7]:=0 end
end ; K:=K+1; go to SW1[K] ;
END1 :
switch SW2:=EXCEP5, EXCEP6, EXCEP7, EXCEP8, END2 ;
K:=1 :
EXCEP5 : for P:=1 step 1 until 6 do ZX[P]:=Z4[P] ; go to KEISA2 ;
EXCEP6 : if W=0 then
begin if XX[5,5]=1 then CH[7]:=1 else CH[7]:=0 ;
if XX[6,5]=1 then CH[8]:=1 else CH[8]:=0; go to END2
end ; for P:=1 step 1 until 6 do ZX[P]:=Z5[P]; go to KEISA2 ;
EXCEP7 : if W=0 then
begin S:=XX[5,5]+XX[5,6]+XX[6,5]+XX[6,6] ;
if S=0 then CH[7]:=0 else CH[7]:=1 ; CH[8]:=0 ; go to END2
end ; for P:=1 step 1 until 6 do ZX[P]:=Z6[P]; go to KEISA2 ;
EXCEP8 : if W=0 then
begin CH[7]:=XX[1,10] ; S:=0 ; J:=10 ; for I:=1 step 1 until 10 do
S:=S+XX[I,J] ; if S $\neq$ 0 then CH[8]:=1 else CH[8]:=0 ; go to END2 ;
end else go to STD ;
KEISA2 : W:=0 ; for P:=1 step 1 until 6 do W:=W+ abs (CH[P]-ZX[P]) ;
K:=K+1 ; go to SW2[K] ;
STD : begin CH[7]:=0 ; CH[8]:=0 end ;
END2 :
MTADDR:=A1 ;
READMT :
Read tape (AA[1:12, 1:10], 72, MTADDR) ;
if AA[1,1]=5 then begin Printstring ["NINSHIKI#FUKANO"] ; go to REDATA ; end ;
I:=11 ; SUM:=0 ; for P:=1 step 1 until 8 do
SUM:=SUM+abs(CH[P]-AA[I,P]) ;
if SUM=0 then
begin code PRINT ; go to REDATA ; end ;
go to READMT ;
END3 :

```


- (1) もっと有効（認識に際して安全）な特徴パラメータを選出する。
- (2) パターンのマトリックスを例えば25×25のように cell の数を増し、精密化する。
- (3) 特徴を算出する方法を改良する。

最後に、本研究をすすめるにあたって、終始ご指導いただいた本学、四谷平治教授に感謝いたします。また、プログラムについていろいろご指示下された本学計算センター石黒さんにお礼申し上げます。

参 照 文 献

- 1) E. A. Feigenbaum et al : Computer and thought (Mcgraw-Hill) 1963.
- 2) R. Casey et al : Recognition of printed chinese characters, IEEE. Trans. EC. Vol. EC-15 No. 1 Feb. 1966, pp. 91~101
- 3) R. Narashimhan : Syntax directed interpretation of classes of pictures. Comm. ACM. Vol. 9. No. 3 March 1966 pp. 166~172
- 4) 飯島泰蔵：パターン認識の理論
電気通信学会雑誌，昭和38年11月，pp. 96~105
- 5) 伊藤貴康，福永圭之介：パターン認識系の反復的構成法，
電気通信学会雑誌，昭和41年10月 pp. 82~88
- 6) 瀬崎信彦他：線がき図形自動追跡装置
エレクトロニクス，昭和41年6月 pp. 1047~1052
- 7) 関口茂：パターン認識
電気通信学会雑誌，昭和41年7月 pp. 113~122
- 8) 坂井利之：文字・図形の認識機械
共立出版，情報科学講座 E. 19. 2 昭和42年
- 9) 雑誌“自然”（人とコンピューター）
昭和41年1月~4月
- 10) 八木，高田，三木：デジタル計算機による文字パターン認識について
昭和42年 電気4学会北陸支部連合大会
(昭和42.11.30 受付)