

調査・製作ワークショップ
「C言語によるパターン認識プログラミング」 荒井 ver.00
第1週&第2週 (本資料は来週も使います)

○パターン認識とは

パターン認識とは、画像処理により対象とする画像に何（パターン）が映っているのかを識別することである。

人間は目を見たものが何であるかを瞬時に判断できるが、コンピュータではこれは難しい。

この授業では、簡単なパターン認識の手法を勉強し、自分で簡単なパターン認識の方法を考えてプログラムする。

○デジタル画像データ

画像データは、画素（ピクセル、ドットと呼ばれることもある）が集まっていて、一つ一つの画素は数値で表現されている。簡単な例で例えるならばオセロ板のようなものだと思えばよい。縦横にマス目（画素）が並び、そこに白か黒のコマを全て敷き詰める。



そこで黒（もしくは白）がどのような形（パターン）であるかをコンピュータにより自動的に調べるのがパターン認識である。

なお、ここでは簡単のため、一つの画素では白か黒のどちらかとし、つまりモノクロ画像を対象とする。ここでは扱わない白黒画像（白と黒の間であるグレーなどを含む画像）やカラー画像などは、一つの画素において、白と黒以外のグレーや各色を表すことができるものである。

例えば、3×3画素の画像データで次のようなパターンの場合、

□□□
■□□
■□■

白を0、黒を1として、

000
110
111

のように表現して扱う。

なお、通常よく使われる画像データは、1600x1200 など非常に多量の画素からなり、かつ各画素において多くの色（16万色以上）が表現できるカラー画像である。

●画像処理

画像から何らかの情報を取り出すためなどで、画像を加工し変形したりする処理のこと。カラーから白黒への変換、モザイク処理、拡大縮小などなど多くの処理が存在する。

○エッジ抽出

特徴の前に、画像処理を体験的にも理解できるよう画像処理の基礎的な加工手法としてエッジ抽出を紹介する。エッジ抽出（エッジ検出）とは、理想的には映っている物体の端（境界）を取り出す処理である。

ここでは、画像データは白黒（0/1）データであるとする。

例えば 7x7 の画像において、次のような画像の黒い画素は四角のパターンと言える。

```
□□□□□□□
□■ ■ ■ ■ ■ □
□■ ■ ■ ■ ■ □
□■ ■ ■ ■ ■ □
□■ ■ ■ ■ ■ □
□■ ■ ■ ■ ■ □
□□□□□□□
```

この場合パターン内部は塗りつぶされているが、周辺のいわゆるエッジ（周辺部分）のみを取り出した方が処理しやすいことがある。

先の画像からエッジを抽出した（理想的な）例を次に示す。

```
□□□□□□□      □□□□□□□
□■ ■ ■ ■ ■ □      □■ ■ ■ ■ ■ □
□■ □ □ □ □ ■ □      □■ □ □ □ □ ■ □
□■ □ □ □ □ ■ □      □■ □ □ □ □ ■ □
□■ □ □ □ □ ■ □      □■ □ □ □ □ ■ □
□■ ■ ■ ■ ■ □      □■ ■ ■ ■ ■ □
□□□□□□□      □□□□□□□
```

○一次微分によるエッジ抽出方法

エッジを抽出する画像処理技法は様々あるが、ここでは、最も簡単な一次微分によるエッジ抽出方法を紹介する。

まず、エッジの意味を考えてみよう。

対象の周辺とも考えることができるが、別の考え方として、エッジは変化する部分であると考えられることもできる。つまり、白□→黒■へ、もしくは逆に変化するところがエッジである。

さて、数学が嫌いな方もいるかもしれないが、変化の具合を示す数学的な道具に、「微分」がある。

微分とは変化の具合のことである。たとえば曲線の接線は、その曲線の傾きを表すことになり、グラフにおいて傾きとはどの程度変化しているかを示していて、傾きは微分で求めることができた。

画像処理における微分は、基本的に引き算である。

例えば、□■（0 1）という画像データの変化具合は、 $0 \rightarrow 1$ と変化しているので、傾きは 1 と言える。□□（0 0）は変化なしで、傾きは 0 と言える。つまり、各々 $(1-0=1)$ 、 $(0-0=0)$ と引き算で求めることができる。これが画像データにおける微分の基本である。

但し、これだけではちょっとまずい。

今、2 x 2 画素の画像を考えて、次の図のように、上二つが白(0)、下二つが黒(1)だとする。この画像の傾きはどうなるであろうか。

```
□□    0 0
■ ■    1 1
```

先ほどのように、横（左から右）の変化を考えると、傾きは

0 (=0-0) ※右の 0 - 左の 0

0 (=1-1) ※右の 1 - 左の 1

となってしまふ。

しかし変化の方向を下向き方向（上から下）で考えると、 $1(=1-0)$ $1(=1-0)$ （※下－上）となる。

そこで、縦横両方の変化具合を合わせた値 d は、次のような計算で求めることがある。

$$\Delta x = f(x+1, y) - f(x, y)$$

$$\Delta y = f(x, y+1) - f(x, y)$$

$$d = \sqrt{\Delta x^2 + \Delta y^2}$$

上記の 2×2 の画像例において、画素(0,0)（左上の画素）に注目した場合、 $\Delta x = 0-0=0$, $\Delta y = 1-0=1$ となり、微分値 d は $1(=\sqrt{0^2 + 1^2})$ となる（C 言語ではルートの計算は `sqrt()` 関数を利用し、二乗は単に同じものを掛ければよい）。

$\begin{array}{cc} \square \rightarrow \square & 0 \rightarrow 0 \\ \downarrow & \downarrow \\ \blacksquare \quad \blacksquare & 1 \quad 1 \end{array}$ 横の変化 Δx は 右－左 = 0
縦の変化 Δy は 下－上 = 1
縦横合わせた変化具合 d は $1(=\sqrt{0^2 + 1^2})$

また、 3×3 の次のパターンの場合における（縦横の）変化具合は、

$\begin{array}{ccc} \square & \square & \square \\ \square & \blacksquare & \blacksquare \\ \square & \blacksquare & \blacksquare \end{array}$ $\Rightarrow$ 変化具合 $\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}$

となる。このように隣（右と下）とのいわゆる差を取るので、画像 3×3 は 2×2 のエッジデータとなる。

また、 3×3 で次のようなパターンの場合、エッジ（一次微分値）を計算し、1 以上を 1 としてエッジ画像を求めると次のようになる。

$\begin{array}{ccc} \square & \square & \square \\ \square & \square & \blacksquare \\ \square & \blacksquare & \blacksquare \end{array}$	$\begin{array}{cc} 0 & 0 \\ 0 & 1.4142 (\sqrt{2}) \end{array}$	$\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array}$	$\begin{array}{cc} \square & \square \\ \square & \blacksquare \end{array}$
原画像	微分値	エッジ（整数）	エッジ画像

なお、画素の値は整数でなければいけないので、たとえば上記ではエッジの計算値が 1 以上の場合は 1 とし、それ以下の場合は 0 とした。

$\Rightarrow$ ★実際に手計算し、それからプログラムしてみよう！

まずは課題用紙の問 1 で手計算をしてみよう。

◎プログラムについて

- ・まず、C 言語における、配列について確認しておこう。
- ・次にサンプルファイルをダウンロードし、実行方法、エラーの見方&修正を。そしてプログラム全体の構造についておよそ理解しておこう。
- それからエッジ抽出処理のプログラムを完成させよう！

画像処理は多量の画素に対して同じ処理をする必要があり、計算が早く繰り返しが得意なコンピュータにとっても有効である。本授業では、実際に C 言語によって、画像処理プログラムを作る。

◆パターン認識の手法概要

パターン認識は、何らかの画像を対象として、それがどのパターンであるのかを識別することである。たとえば、先のおセロで作ったものを見て、「ハート型」とであると判定することである。

パターンを認識もしくは識別するには、様々な方法がある。

まず、考え方としては非常に単純な方法が、「**テンプレートマッチング**」である。これは、識別すべき物体のサンプル画像を切り抜いてテンプレートとし、認識をしたい画像に当てはめていき、ぴったりと重なったら識別できたとする方法である。

考え方はシンプルだが、実際には様々な問題がある。

まず、通常「ぴったりと重なる」つまり全く同じパターンが現れるとは限らない。これはノイズ（雑音）が生じたり、観測誤差が生じたりすることもあるばかりか、実際パターンが全く同じとは限らない。更に大きさが違ったり、回転されていたりすることもある。

もう一つのパターン認識の大きな考え方は特徴などによる「構造マッチング」である。この方法はテンプレートマッチングが、画像そのものを照合させていくのに対し、面積などのパターンの特徴を抽出しておき、これらを照合していく考え方である。

パターン認識においては、どのような形（パターン）でも認識できるようにすることはほぼ不可能である。たとえば、●と×の2種類のパターン（形状）を認識したいとした場合、★のようなパターンは認識できなくても問題ない。この場合、●と認識しても、またどちらでもないと認識しても構わない。

さて面積という特徴では大きさしかわからないので、必ずしも形がわかるわけではない。しかし、ここでは対象とするパターンは、ほぼ決められた一定の大きさであるとする。このような条件を付けければ、たとえば、●と×というパターンを考えた場合、明らかに面積は違う。つまり、認識したい対象の画像において、面積を観測し、その面積が大きければ●、小さければ×と判定することができる。具体的に考えると、たとえば●の面積はおおよそ 220 程度だと考えられ、×の面積はおおよそ 60 程度だと考えられる場合、あるパターンの面積を観測した結果、180 だとしたら、そのパターンは●であると判定させればよい。より具体的には、たとえば●の面積は 200～240 で、×は 30～90 とし、それ以外の面積の場合はこの 2 種類のパターン以外であると判定させるなどとする。

○特徴量

特徴としてよく利用される幾何学的特徴としては、面積以外にも「密度」、「周囲長」や「アスペクト比（縦横比；縦の長さと横の長さの比）など様々なものがある。通常雑音除去や二値化、エッジ抽出などの画像処理をした上で計測する。

◇面積

パターンの面積のことである。

ここではパターン（物体）は、黒(1)で表現することになっているので、単純に全画素の値を合計すれば簡単に求めることができる。

例えば、次のような 2x2 画素のパターンの場合、

□	■	0	1
■	■	1	1

黒の画素は全部で三つなので、面積は 3 となる。これをプログラムで実現するには、 $0+1+1+1$ というように全画素を合計すればよい。

⇒★実際のプログラムで確かめてみよう！

面積を求めるプログラムをサンプルプログラムで確認しよう。ただし、上半分と下半分の面積（下記参照）を各々抽出して合計しています。

◇上下の面積比

面積を対象画像の上半分と下半分で、各々抽出し、それら面積もしくは面積比を特徴とする。

つまり全体の面積がもし同じ程度だとしても、上半分の面積もしくは下半分の面積で比べると違いが生じることもまる。また、上下に対象なパターンは、面積比はほぼ 1.0 になり、上下に対象ではないパターン（たとえば▲など）は面積比は 1.0 とは異なる。

ここでいう面積比とは、上半分の面積を S_u 、下半分の面積を S_d とした時、 S_u/S_d で計算した値のことをいう。

例えば、次のような 2x2 画素のパターンの場合、

□	■	0	1
■	■	1	1

上部分の黒の画素は 1 個で、下部分では二つなので、上下それぞれ面積は、 $1 (=0+1)$ 、 $2 (=1+1)$

となる。この場合の上下の面積比は、上の面積÷下の面積=1/2=0.5 となる。

よって三角形▲のような物体であれば、値は小さくなり、▼だと大きくなる。また◆のようなものだとほぼ 1.0 になるはずである。

◇周囲長

周囲長とは、そのパターン（物体）の周囲の長さのことである。

周囲長を求めるにもさまざまな方法が考えられる。一般的には、パターンの周囲を追跡して求め、このつながりの距離を求める。なお、斜めにつながっている場合は、距離は1ではなく、 $\sqrt{2}=1.4142$ とすることが多い。

しかし、少し考えてみると、周囲長をもう少し簡単に求めることができる。

今我々はエッジ抽出をすることができる。エッジも基本的にはパターンの周囲とほぼ同等である。よって、エッジ抽出をしたエッジ画像において、エッジ部分の面積を求めればよいことになる。なお、実際に計測してみるとわかるが、必ずしも正確なエッジではなく、かつ画素数が少ないこともあり、周囲長は▲、●、◆を識別するための良い特徴ではない。

◇ヒストグラム（度数分布）

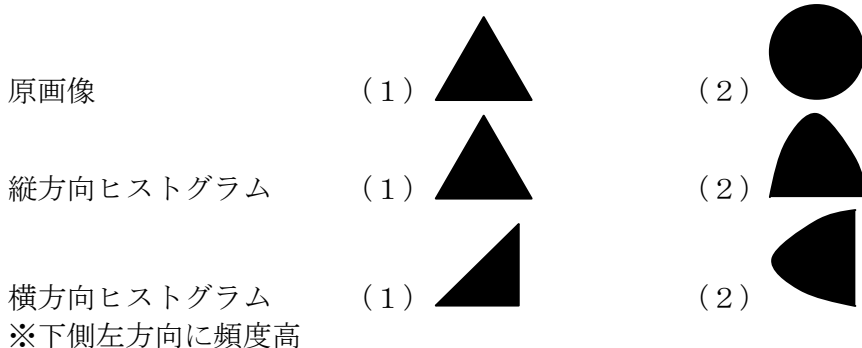
画像データは、二次元なので、ヒストグラムは縦方向と横方向の二つを求めることができる。ここではパターン（物体）は黒(1)であるので、たとえば縦方向のヒストグラムの場合、ある列（縦一列）において、黒の画素の数を数えることにより求められる。

ヒストグラムはこのままでは特徴としては使い難い。これは単に二次元データが一次元になっただけであるので、二つの一次元データを比べるのが難しいからである。

ではどのようにヒストグラムを利用すればよいであろうか？

例えば、今、▲、●の2つのパターンを識別することを考える。縦方向のヒストグラムを考えると、▲も●もどちらも中膨れした形状（ほぼ▲）になるはずで、識別は難しそうである。

では横方向のヒストグラムを考えてみよう。▲は、上が小さく&下が大きくなり（∟のような形状）、●は、中膨れの形状になる。



よって、大きく3つに分けて考えればよい。つまり、縦方向のヒストグラムの場合は左・中・右部分、横方向のヒストグラムの場合は上・中・下部分に分ける。そして各々の部分において度数がおおよそ小さいのか、中くらいなのか、大きいのかを判断する。なお部分に分けるには、簡単に単にそれらの部分の合計を適当に判断させてもそれなりの精度が出せる。

より具体的に考えると、たとえばヒストグラムの上部分とは、 $i=1\sim5$ の行とし、その5行の度数合計が10以下ならば小、10~60ならば中、60~100ならば大などと判定することが考えられる。なお、この数値（判定基準値）は、数学的かつ統計的に求めるべきであるが、ここではアルゴリズム作成者（つまり自分自身）が十分に考えた上で妥当かつ適切に決めればよい。いずれにしても「上部分」や「下部分」や、「大きい」とか「小さい」といった定性的な表現を、具体的にどのようにするかを決めたり、数値に置き換えたりする必要がある。

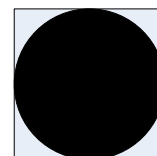
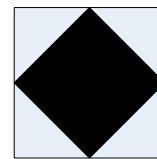
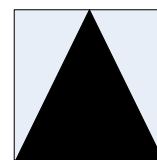
次に今度は、▲、●、◆の三つのパターンを識別することを考える。残念ながら●、◆は、縦方向・横方向共にヒストグラムで区別するのは難しそうであり、別の特徴で区別すべきである。

○パターン認識の例；特徴の選択

面積という特徴について、どのようなパターンが識別できるかを考えてみよう。

認識したいパターンを▲と◆の2種類として考えてみよう。

今、正方形の画像データ中に、ほぼ一杯にほぼ正確なパターンが現れると仮定してみよう。



う。簡単な算数の問題であるが、このような理想的なパターンだとすれば、▲も◆も、どちらも面積は同じで、20x20画素の正方形の画像とした場合、どちらも面積は200となる。

よって、▲と◆は、面積という特徴では、区別はできないことになる。

では、どのような特徴だったらこの二つを区別することができるだろうか？

次に、◆と●の2種類のパターンについて考えてみよう。

このような理想的な（正確な）パターンの場合、●の面積は円の面積の公式(πr^2)から約314となるはずで、◆の200よりも大きい。

理想的ではない現実的なパターンは、ノイズや誤差等があるので、これを十分に考慮して考えなければいけない。例えば、◆の面積は180~220、●の面積は「280~340」などと想定することが考えられる。これらの値は、必ずしも正確に決めることはできない。が、十分考慮して理由ある値として設定すべきである。

◇実際のパターン認識

パターン認識をする場合、実際には一つの特徴を使うわけではない。つまり、複数の特徴を使用し、これらを総合的に判断してどのパターンかを識別する。

ここでは、▲、●、◆の画像パターンを識別することを考えてみよう。また、各々のパターンは必ずしも正確な図形ではなく、また雑音も生じているとし、実際にパターン認識の方法を考えてみよう。つまり、単純なテンプレートマッチングでは難しく、また面積特徴だけでは全てのパターンを識別することは難しい。また、ヒストグラムだけでも難しい。

ここまでで、面積及び上下部分の面積（比）とヒストグラム（{原画像, エッジ画像}×{縦, 横}の4種類）を特徴として利用することができる（ここでは周囲長は良い特徴ではないので利用しないとする）。但し、ヒストグラムは、このままではなく、上記の様に更に工夫して利用する必要がある。

では、どうしたらよいであろうか？いずれも単独の特徴量ではうまくいかないのが、複数の特徴をうまく組み合わせればある程度解決可能である。例えば▲パターンの場合の各特徴を考えてみると、面積は中くらいで、原画像の縦方向のヒストグラムが左から小・大・小で、横方向が上から小・中・大、エッジ画像の縦方向ヒストグラムは小・小・小、横は小・小・大(中)となるはずである。このように考えて▲、●、◆を区別できる特徴を組み合わせればよい。

なお考え方、つまりパターン認識の方法の正解は一つではなく、複数存在する。但し絶対完璧な認識ができる方法はほぼ困難である。また例えば大中小などの判断なども必ずしも正解があるわけではない。より柔軟に様々な変形パターンがなるべく正確に認識できるように考えてみよう。人間でも●だか◆だか判別できないパターンもあり、100%識別するパターン認識アルゴリズムはほぼ困難（不可能に近い）であることは十分に頭に入れておこう。

特徴によるパターン認識の方法を考える場合、次の点に注意しておこう。

複数のテスト画像を同時に比べるのではなく、ある一つの画像がどのパターンなのかを調べるようにすることに注意しよう。たとえば、▲画像と●画像を比べて、どちらが▲パターンでどちらが●パターンなのかを調べたいわけではない。あくまでも一つの画像を対象に▲パターンなのか●パターンなのかを認識したいわけである。

○パターン認識の方法

パターン認識するには識別したいパターンの特徴を把握しておく必要がある。たとえば、パターンAは面積が大きく&周囲長が短い、またパターンBは面積が小さく&周囲長が長いという特徴があれば、識別したいテスト画像の面積と周囲長を観測して、どちらのパターンかを決定させる。なお、この場合、どちらのパターンでもない、例えば面積が小さく&周囲長も短ければ、どちらのパターンでもないとするのが自然である。但し、テストパターンとしてAかBのどちらかしか出現しないことが分かっている場合は、どちらに近いかで判定する。

・三つのパターンの識別！

パターン A▲、パターン B●、パターン C◆の3つのパターンを識別する方法を考えよう。

通常一つの特徴で3つを区別するのは難しい。例えば面積だと、少なくとも▲と◆は区別できないことは説明したとおりである。では、そのように三つを識別していったらよいだろうか。

通常ある一つある一つの特徴により、二つに分類することを考える。例えば、1：▲と◆、2：●などのように。そして次に▲と◆を区別する特徴を加え、ツリー（木）構造的に判別させていくとよい。つまり、これら三つのパターンの特徴をよく考え、まず二つと一つに分けやすい特徴が何かを考えていくことになる。微妙に識別できるのではなく、なるべく明確に識別できる特徴を探す。それから二つのパターンを明確に区別できそうな特徴を探す。

例えば、今三つのパターンのいくつかの特徴は次のようになっている（一部省略）。

パターン	原画像	原画縦 Hist	原画横 Hist	エッジ画像	エッジ縦 Hist	エッジ横 Hist
A	▲	▲ 小大小	▲ 小 中大 大	△	— 小小小	└ 小 小 大
B	●	● 小	● 小	○	— 小小小	└ 小 小 小
C	◆	▲ 小大小	▲ 小 大 小	◇	— 小小小	└ 小 小 小

この時、どの特徴を用いると、A、{B, C} もしくは {A, B}, Cまたは {A, C}, Bに分けやすいのかを考えればよい。

上表でみると、少なくともエッジ縦ヒストグラムはこの三つを区別するには、全く役に立たない特徴であることが分かる。但し考え方によっては、この三つのパターンのエッジ縦ヒストグラムは必ずこうなるということも言える。

上表には示していないが、面積などを含めた全ての特徴についても検討していく必要がある。それらを総合して、まずどの特徴を用いて二つの組に分け、次に二つのパターンをどの特徴で区別するのかを考えていけばよい。

ちなみに上表にはないが、周囲長は、今回のプログラムでは結構誤差が大きいことにも注意が必要である。微妙な判定はできないと思った方がよい。

○より工夫を！より柔軟で正確な識別を！！

このようにどのような特徴を使うかは非常に重要である。

上記では、エッジ横ヒストにより、Aと{B,C}を分け、次にBとCを分けるのに面積を利用することを考えた。しかし、面積はBとCを区別するには実際は微妙かもしれない。理想的には確かにB●の面積はそこそこ大きいはずが、ノイズや誤差があると微妙となってくる。よって、BとCをより明確に区別する特徴がないかを考えるとよい。

例えば、ヒストグラムは縦方向と横方向しか考えていなかったが、斜め方向を考えてみるのも手である。エッジ画像の斜め方向のヒストグラムは、BとCを区別するのにかなり有効な特徴であることがわかる。

また、3つのパターンをよくみると、B,Cは点対称な図形であるのに対し、Aだけは点対称ではない。これを利用するにはどうしたらよいであろうか？図形の偏り具合を数学的に表現することも十分に可能であるが、ここでは面積を少し工夫して考えてみよう。たとえば、画像を上半分と下半分に分けて考える。パターン A▲は、上半分の面積は小さく下半分は大きい、一方パターン B●も C◆も、上半分と下半分の面積はほぼ同じである。つまり、上面積/下面積を計算すると、パターン Aは小さな数字（0.5 程度？）となり、パターン B,C はほぼ 1.0 になるはずである。

勿論この上下面積比も万能ではない。今、パターンは回転を許さず、■はパターン C◆とは違うパターンであるとした場合は、この上下面積比は有効であるが、回転を許し、■もパターン C◆の

変形であるとした場合には、この特徴は利用できなくなる。

次に、今回は3つのパターン▲●◆を対象とはしているが、例えばごみのような「・」や「／」が識別となってしまった場合にでも、▲●◆のどれかに識別されてします。これは明らかにおかしいので、あらかじめ設定されている3つのパターン以外であると拒否（リジェクト）してしまうとよい。

いずれにしても、様々な変形・回転や雑音などを許して、柔軟にパターン識別させることは、非常に難しく、人間のパターン識別能力の素晴らしさには脱帽せざるを得ない。近年パターン識別、特に一般物体識別に対する新たな効果的手法が開発されて盛んに研究&実用化されてきた。たとえば、「一般的な物体」として「人の顔」を検出するデジカメなどはかなり普及してきている。それでも識別・抽出の正確さ、識別処理時間や、学習数、学習時間などなど様々な面で問題点は残されており、コンピュータの目が、人間に近づくにはまだまだ遠い未来と言えよう。

一方で、難しそうなことも、ちょっとした工夫によって簡単な方法で解決できることがあることも知っておこう。

※よりよいパターン認識への工夫（まとめ）

今回のテストパターンは例えば▲で一つしかないが、様々な画像を柔軟かつ正確に認識できるように十分工夫しよう。

また、パターンの回転は基本的に考えておらず、三角は▲のみを想定し、▼は識別対象外としていた。しかし、これも可能ならば両方とも、更に▶なども三角として認識できるように考えてみよう。

また、今回は3つのマーク（パターン）を識別するものであるが、全く違う（例えばごみ）を観測してしまっても、3つのどれかに識別されてしまう。よって、これは明らかに設定された3つのパターンとは違うとして、はじいてしまうことが良い。これをリジェクトという。

時間があれば、どんどんとこれらの処理を考えてプログラムしてみよう。

★次週までの宿題！

- ・各特徴の抽出プログラムを完成させておくこと！

また、実際にどのような数値になるのか確かめておいてください。

ちなみにサンプルプログラムでは、上記で紹介した特徴について、およそ完成しています。未完成な部分としては、面積抽出部分の文法エラー、エッジ抽出、縦横のヒストグラム抽出、3分割ヒストグラムの大中小の判断が未完成です。特徴の抽出プログラムのこれらを完成させて、▲パターンらしき画像、●パターンらしき画像、◆らしき画像の各々に対して、実際の各特徴の値がどのようなようになるのかを確かめよう。

なお、サンプルプログラムでは、20x20の画像の中に、およそ18x18に収まるパターンとされていることに注意しておこう。

- ・パターン認識の方法を考えてきてください！

今週は画像処理部分でしたが、次週パターン認識部分を各自で作成して頂きます。

認識方法は、人によって違います。各自で考えて、その方法をプログラムして頂きます。

但し次週は考えている時間はあまりないので、必ず方法（どの特徴を利用し、どのように認識させるのか）を考えてきてください！

次週はパターン認識部分を含めたプログラム全体の完成と、レポートの作成を行います。