



平成 23 年度 卒業論文
観光用リアルタイム画像認識システムの構築

釧路工業高等専門学校

情報工学科

兒玉拓巳

2012 年 3 月

目 次

第1章 はじめに	1
1.1 研究背景	1
1.2 画像認識とは	2
1.3 システムのインターフェース	2
1.4 リアルタイム識別	3
1.5 画像認識技術の発展	3
第2章 システムの概要	4
2.1 OpenCV ライブライ	4
2.2 システム開発環境	5
2.3 品種特定までの流れ	5
2.4 第1フェーズ カメラ映像の処理	6
2.4.1 カメラ映像の入力	7
2.4.2 花弁色の指定	8
2.4.3 閾値処理による花弁領域の分離抽出	9
2.4.4 マスク画像の作成	10
2.4.5 ノイズの除去	11
2.4.6 ラベリング処理	12
2.5 第2フェーズ 映像から各特徴の抽出	13
2.5.1 色特徴 $\overline{H}, \overline{S}, \overline{V}$ の抽出	14

2.5.2	形状特徴 \bar{C} の抽出	15
2.5.3	距離特徴 \bar{D} の抽出	16
2.5.4	特徴抽出のまとめ	17
2.6	第3フェーズ 特徴量データを用いた品種の識別	18
2.6.1	k 最近傍決定則	19
2.6.2	学習パターン群の生成	20
2.6.3	機械学習を用いた識別	21
2.7	システム稼動中の様子	22
第3章	評価実験	23
3.1	評価実験: 概要	23
3.2	評価実験: 結果	24
3.3	評価実験: 考察	25
3.4	大規模クラス識別実験	26
第4章	今後の改善点と課題	27
4.1	現状における当システムの問題点	27
4.2	今後増やしたい機能・装置	28
4.3	社会的貢献	30
第5章	おわりに	31
第6章	謝辞	32
参考文献		33
発表論文・その他		34

図 目 次

1.1	品種特定までの流れ	2
2.1	システムへの入力	7
2.2	色のサンプル	8
2.3	閾値処理の流れ	9
2.4	マスク画像作成の流れ	10
2.5	処理前と処理後の画像の比較	11
2.6	ラベリング処理済の画像	12
2.7	実際の抽出色	14
2.8	凸包を用いた割合値の計算イメージ	15
2.9	距離特徴を抽出するまでの流れ	16
2.10	参考入力パターン	17
2.11	入力パターンから特徴空間への写像	18
2.12	k 最近傍決定則のイメージ(仮)	19
2.13	教師ラベルを付加した学習パターン群	20
2.14	各クラスの学習パターンの内訳	20
2.15	リアルタイム画像認識のイメージ	21
2.16	システム稼動中の様子	22
3.1	大規模クラス識別実験に用いた 8 クラス	26

4.1 礼文島の位置	30
----------------------	----

表 目 次

2.1 開発環境	5
2.2 閾値一覧	8
2.3 抽出した特徴量データ一覧	17
3.1 評価実験の対象クラス一覧	24
3.2 3 クラスの識別結果	25
3.3 8 クラスの識別結果	26

第1章

はじめに

本章ではまず研究の目的や背景、タイトルの意味、画像認識技術について述べる。

1.1 研究背景

旅先や外出中にて道端にて花を見つけた際、その花の色や形などが分かっても品種までは分からないことが多い。品種名を知りたければ、図鑑や Web を用いて手探りで特定するしか方法が無い。とはいえ、自ら図鑑を広げてまで品種を特定しようとする人はまず少なく、大抵の人は撮影した画像と共に「あ、綺麗な花だ」という感想だけで終わってしまうだろう。

そこで、本研究では近年急激に進歩する画像認識技術を用いて、画像データから簡単に花の品種名を特定するシステムの構築を提案する。識別対象は北海道に咲く花の画像とし、画像データ内の花の色や形状などの特徴から、コンピュータが自ら花の品種を識別可能となることを目標とした。

1.2 画像認識とは

現在では画像認識という言葉自体が広く用いられているが、本研究における画像認識の概念は画像から対象物の特徴を抽出¹し、その特徴から画像内の対象物が何であるのかを判断することである。

この概念を花を例として説明すると、花の品種を見分ける際に重要となる点(花弁の色や形など)を特徴と呼び、特徴を挙げることを特徴を抽出する、という。そして挙げられた複数の特徴を吟味し、その花の品種が何であるのか判断までを行うことを画像認識と定義している。

これらの作業は、人間は普段から無意識に行っている行動であるが、コンピュータに判断させると非常に高度で複雑な処理となることが知られている。

1.3 システムのインターフェース

本研究はシステムの構築というタイトルのため、具体的にどのようなシステムにするかを述べておく。



図1.1 品種特定までの流れ

利用者は、パソコンに接続されたウェブカメラに対して撮影した花の画像を提示するだけで作業は終了する。その後は当システム内で「映像内に何の花が映っているのか？」を画像認識を用いて自動的に判断、そして結果の表示を行う。

¹特徴抽出方法の詳細などは第2章で述べる。

1.4 リアルタイム識別

当研究ではただ単に画像に対して識別を行うだけでなく、リアルタイムで入力されるカメラ映像に対しての識別を行う。

まず、利用者がウェブカメラに対して花の画像を提示すると、当システム内ではカメラ映像に写る花の画像内の特徴をリアルタイムに抽出し続ける。次に、その抽出された情報に対して、同時に機械学習を連動させることにより、今現在映っている映像と共に識別を行うことができるリアルタイム識別を可能とした²。

この機能を実装することにより、利用者にとって簡単で分かりやすく撮影した花の品種を特定できる環境を構築した。

1.5 画像認識技術の発展

このシステムのように、画像認識技術を用いて、ある分野における膨大なオブジェクト集合の中から一種類を気軽に特定できるようなシステムが普及すれば、人々はその分野に対しての知識や興味がより深まることが予想される。

分野として挙げられるものは、主に身近に存在しながらも詳しくは知られていないものが良いと考えられる。それは花の品種名に限らず、野鳥の鳥名、自動車の車種名、道路標識の名前など挙げてみると多々存在する。画像認識技術は人々がこれらの分野を深く知るための第一歩と成り得るため、人の興味と深い知識の橋渡しとなれるようなシステムの構築を試みた。

²より技術的な説明は2.6.3節に掲載している。

第2章 システムの概要

本章では開発環境やシステムの処理の流れについて述べ、実際に構築方法を順を追って解説してゆく。

2.1 OpenCV ライブライ

当システムの開発には Intel 社が開発した OpenCV ライブライを用いた。OpenCV を採用した理由は以下の通りである。

- カメラライブライとして有名でカメラの導入も容易であり、機能が豊富。
- クロスプラットフォームであり、言語も C/C++ と Python に対応している。
- 機械学習関数が豊富で、特徴抽出後に識別まで行うことができる。

つまり、OpenCV ライブライのみでカメラ映像の処理から機械学習で識別を行う部分まで実装することができる。そのため、外部プログラムに頼る必要性がなく、全て 1 つのプログラムで完成させることができた。

2.2 システム開発環境

当方の開発環境を表2.1に記しておく。こちらも比較的揃えやすい環境で開発を行うことができた。

表2.1 開発環境

CPU	Intel Core 2 Duo
メモリ	4GB
OS	Windows 7
IDE	Microsoft Visual C++ 2008

また、OpenCVのバージョンは2.0を使用。ウェブカメラはLogicool社製の市販のものを使用した。

2.3 品種特定までの流れ

本システムを用いて花の品種を特定するにあたって、作業を大きく3つに分けることができる。ここではそれぞれの作業をフェーズと呼び、分割して解説を進める。

第1フェーズ ウェブカメラから映像を入力し、閾値処理により花弁領域を抽出する。また、第2フェーズにおいて特徴抽出の精度を上げるためにノイズ除去やラベリング処理といった映像編集を行う。

第2フェーズ 第1フェーズで編集された映像に対して、花弁の平均色・形状などの特徴を抽出。抽出された情報を特徴量データへと変換する。

第3フェーズ 第2フェーズで得られた特徴量データを基に、機械学習を用いて識別を行う。識別結果が出次第、品種を特定し結果の表示を行う。

以上の3フェーズを行い、映像内の花の品種を特定する。

2.4 第1フェーズ カメラ映像の処理

本節以降は2.3節に示した各フェーズごとの解説を行うこととする。まず第1フェーズではカメラ映像を入力後、以下のような編集を行う。

- 品種を知りたい花の花弁色をユーザが指定し、指定色をターゲット色とする。
- 閾値処理によりターゲット色を抜き出し、花弁領域の分離抽出を行う。
- 分離抽出後の映像のノイズを除去し、ラベリング処理を施す。

閾値処理には HSV 表色系を用い、*H*(*Hue*: 色合い)、*S*(*Saturation*: 鮮やかさ)、*V*(*Value*: 明るさ) の 3 値を閾値で区切って処理をする。

表色系の代表的なものとして RGB 表色系があるが、HSV 表色系は RGB 表色系から変換を行うだけで使用することができ、RGB 表色系と比較して画像内の領域を分離抽出しやすいというメリットがある。

2.4.1 カメラ映像の入力

まず最初に、識別対象となる花の画像(以下、入力パターン)をシステムへと入力する。方法は至って簡単で、入力パターンを webcam へとかざし、読み込ませるだけで完了する。

また、システムへと入力させるために webcam にかざす媒体は、デジタルカメラのディスプレイやスマートフォン・タブレット端末のディスプレイから、印刷されたL判写真にまで幅広く対応している。このため、気軽に撮影した花の写真も一種の入力パターンとなり、簡単にシステムを利用することができる。



図 2.1 システムへの入力

図 2.1 でも示している通り、入力パターンの画像を表示できるものなら何でも入力媒体として扱うことができる。

2.4.2 花弁色の指定

次に、システム内においてユーザに入力パターンの花弁色を指定してもらう。これは、花弁色を指定することにより閾値処理を行い、花弁領域のみを分離抽出するという目的がある。色は三原色に基づいて図2.2に示す6色用意し、各閾値は表2.2のように設定した。

図2.2 色のサンプル

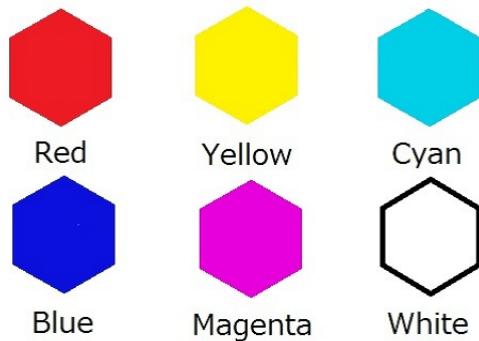


表2.2 閾値一覧

色	$H[0, 180]$	$S[0, 255]$	$V[0, 255]$
赤	$0 \leq h < 15 / 165 \leq h < 180$	$76 \leq s < 255$	$127 \leq v < 255$
黄	$15 \leq h < 40$	$76 \leq s < 255$	$127 \leq v < 255$
シアン	$75 \leq h < 105$	$76 \leq s < 255$	$127 \leq v < 255$
青	$105 \leq h < 135$	$76 \leq s < 255$	$127 \leq v < 255$
マゼンタ	$135 \leq h < 165$	$76 \leq s < 255$	$127 \leq v < 255$
白	$0 \leq h < 180$	$0 \leq s < 128$	$127 \leq v < 255$

ここで指定された色をターゲット色として入力パターンの閾値処理を開始する。ターゲット色を決定する際は H, S, V それぞれの値の範囲を $[0, 180], [0, 255], [0, 255]$ として扱った。

2.4.3 閾値処理による花弁領域の分離抽出

次に、入力パターンに閾値処理を施し、花弁領域のみを分離抽出する。閾値処理とは、画像内にて指定した各 H, S, V の閾値を満たしているピクセルのみを抽出し、2 値化(閾値を満たしていれば白、満たしていないければ黒)を行うものである。ターゲット色の取得までは以下の手順で行う。

1. 2.4.2 節で指定したターゲット色より H, S, V の各閾値を決定する。
2. 画像を H 領域、 S 領域、 V 領域の 3 つに分割し、それぞれ閾値を満たしているピクセルのみを白 (=1)、満たしていないピクセルを黒 (=0) とする。
3. H 領域、 S 領域、 V 領域の各 2 値化画像の論理積(AND)をとり、1 枚の画像に合成する。

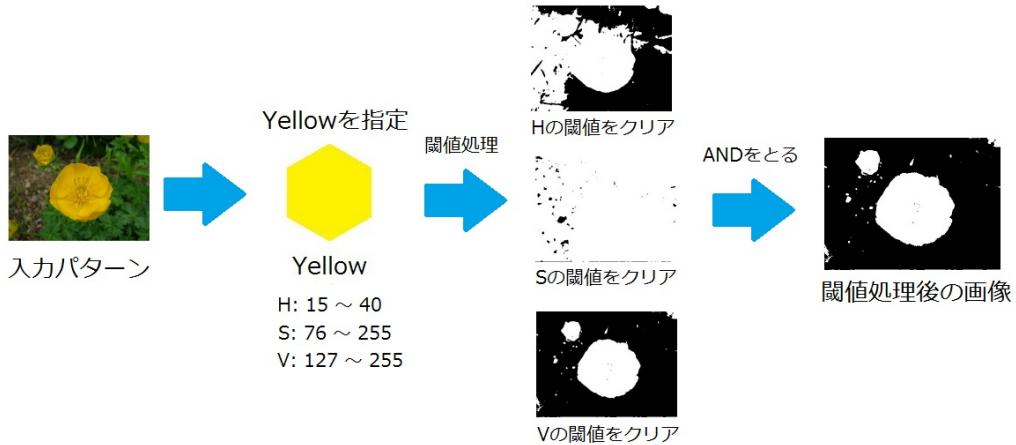


図 2.3 閾値処理の流れ

図 2.3 を例にとると、まず入力パターンの花弁色は黄色のため黄色を指定し、閾値処理で画像を H 領域、 S 領域、 V 領域の 3 枚に分割する。次にこの 3 枚の 2 値画像の AND をとると、指定色の閾値領域のみを分離抽出することができる。

2.4.4 マスク画像の作成

先ほど花弁領域のみを分離抽出したと述べたが、本当にターゲット色が上手く抽出できているかは少々疑問である。そこで、本システムでは確認のためにも、マスク画像を作成して花弁領域を分離抽出できているかを確かめることとした。

作業は簡単で、2.4.3節にてターゲット色の分離抽出が完了した2値画像をマスクと置き、元画像のうちマスクが白(=1)である領域のみを抜き出して表示を行うだけで済む。

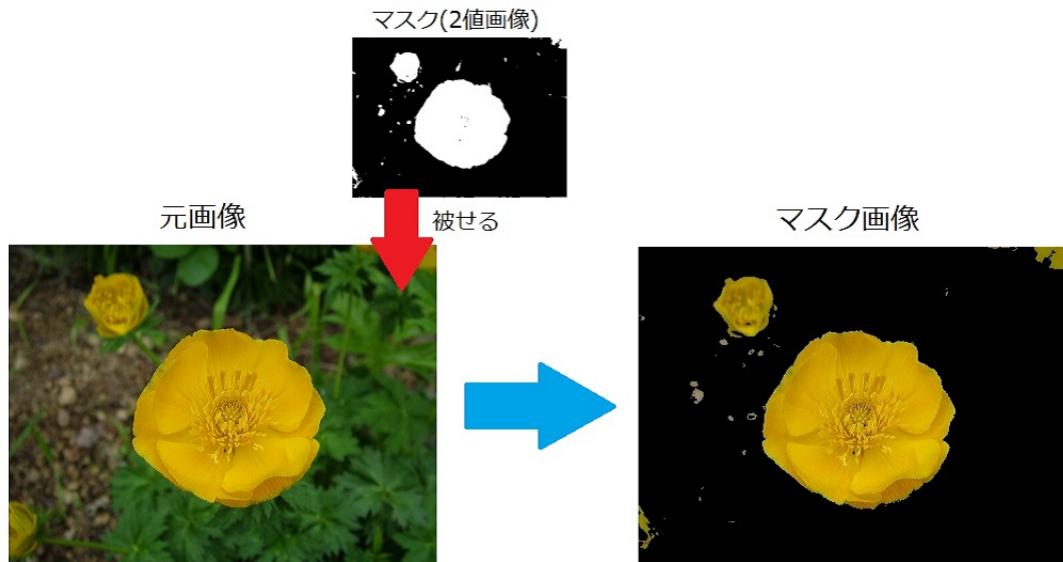


図 2.4 マスク画像作成の流れ

図 2.4 にてマスク画像作成の流れを示している。元画像に対してマスクを被せる(正確には、2値画像が白の部分のみ表示を行う)と、マスク画像を作成することができる。マスク画像ではターゲット色を綺麗に分離抽出できていることが確認できた。

2.4.5 ノイズの除去

次にノイズの除去を行う。ノイズの除去は、閾値処理の段階で誤って取得してしまった葉の部分や、他の小さな花の写りこみを除去し、識別精度や処理速度を向上させる目的がある。また、花弁領域内にて欠損が生じていた場合も、欠損を埋めることができる。

除去方法は、画像の圧縮と膨張を繰り返すことによって細かなノイズを潰す手法を採用している。本システムにおける圧縮・膨張の回数はそれぞれ2回ずつであるが、処理は関数化しているため回数の指定は可能である。一般に、回数が少なすぎればノイズの除去は効果が得られず、多すぎると元画像の形が崩れてしまうという欠点がある。

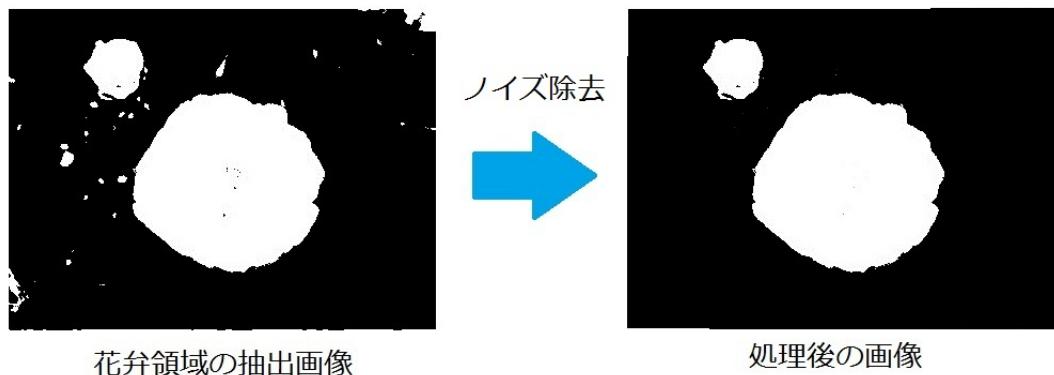


図 2.5 処理前と処理後の画像の比較

図 2.5 で示す通り、ノイズ除去によって細かな点が圧縮と膨張によって消え、花弁内の欠損領域も補完できていることが分かる。また、花弁の形状はノイズ除去前から変わりが無いことも分かる。

2.4.6 ラベリング処理

ノイズの除去が完了したら、続いてラベリング処理も行う。ラベリング処理は、ノイズ除去では消えなかった入力パターン内の花弁領域に対して、大きいものから昇順に順位(ラベル)をつけていく処理である。

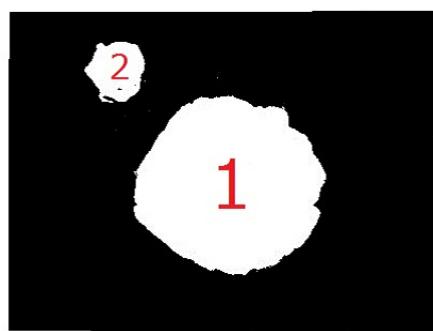


図2.6 ラベリング処理済の画像

図2.6は2.4.5節で作成されたノイズ除去後の画像に対してラベリング処理を施した結果である¹。この例ではラベリング処理の対象となる花弁は2枚しかなかったため数値も1と2のみであるが、更に複数枚花弁が撮影されている入力パターンの場合は数値も1,2,3,...と増加していく。

本研究では、対象とする花弁は1クラスタで十分であるので、ラベルが1の花弁のみを識別の対象とする。もちろん、識別対象ラベルを2や3まで増加させ、複数クラスタの花弁を識別対象とすることも可能である。

この処理を行うことにより、入力パターン内で一番大きな花弁を識別対象とすることができるため、より正確な識別を期待することができる。

以上で入力パターンの画像編集は終了し、次節からは編集した画像から実際に各特徴を抽出する。

¹画像内にある1と2の数値は例として示しているだけであり、実際の画像には書き込まれない。

2.5 第2フェーズ 映像から各特徴の抽出

映像内から識別対象となる花画像データの特徴を抽出する。ここで抽出する特徴は以下の5特徴である。

抽出する特徴

色特徴 \bar{H} 花弁上の H (色相) の全ピクセル平均値

色特徴 \bar{S} 花弁上の S (彩度) の全ピクセル平均値

色特徴 \bar{V} 花弁上の V (明度) の全ピクセル平均値

形状特徴 \bar{C} 凸包 (*ConvexHull*) 内の花弁面積の割合

距離特徴 \bar{D} 花弁の重心 G から各頂点までの平均距離値

各特徴の抽出方法は次節から解説する。

2.5.1 色特徴 $\bar{H}, \bar{S}, \bar{V}$ の抽出

編集を終えた入力パターンを対象に、入力パターンの実際の花弁色を調べる。この色は初めに指定したターゲット色とは異なり、実際の花弁の平均色を調べて色特徴として抽出する。特徴抽出のアルゴリズムを以下に示す。

- 1 ラベリング処理済みの入力パターンの全ピクセルを走査する。
- 2-1 走査上で、もし2値画像のピクセルが黒($=0$)ならば、花弁領域ではないので何もしない。
- 2-2 もし2値画像のピクセルが白($=1$)ならば、花弁領域であるので、変数 $count$ に1を足し、そのピクセルの H, S, V の値を変数 h, s, v へそれぞれ加算してゆく。
- 3 全ピクセルの走査が終えた後、 h, s, v をそれぞれ $count$ の値で割ると、花弁領域上の H, S, V の平均値が出てくる。

以上の作業で出てきた H, S, V の各平均値をそれぞれ色特徴 $\bar{H}, \bar{S}, \bar{V}$ として抽出する。実際にこのアルゴリズムを用いて求めた平均花弁色を図2.7に示す。

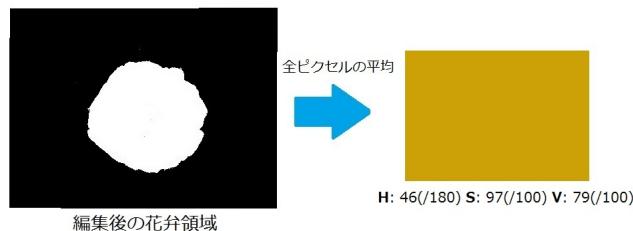


図2.7 実際の抽出色

各特徴は $\bar{H}: 46, \bar{S}: 97, \bar{V}: 79$ となり、平均すると見た目よりも少し暗い色が抽出された。ここで、ターゲット色を決める際の S, V の値はそれぞれ上限が255であったが、平均花弁色を求める際には正規化を行い上限をそれぞれ100と設定している²。

²HSV表色系では色を表現する際に S, V の上限を100%として百分率で表すことが多いため。

2.5.2 形状特徴 \bar{C} の抽出

色を抽出し終えた後は、花弁の形状特徴を抽出する。ここで扱う形状特徴とは、凸包 (*ConvexHull*) を用いて花弁面積の割合 ($0.0 < c \leq 1.0$) を計算した値のことである。特徴抽出のアルゴリズムも含め、以下で解説を行う。

凸包 (*ConvexHull*) とは

ある図形を含む最小の凸図形のことであり、割合値の計算は以下の手順で行う。

- 1 花弁の凸である部分を輪ゴムをかけるように線で囲い、凸包領域とする。
- 2 凸包領域、花弁領域の面積をそれぞれ計算し、値を出す。
- 3 凸包領域内における花弁面積の割合を求め、割合値 c を出す。

以上の作業で計算された凸包の割合値 c を形状特徴 \bar{C} として抽出する。

実際にこのアルゴリズムで求められた凸包の割合を図2.8に示す。この例では、赤い点線で囲まれた部分(凸包領域)に対して、花弁面積の割合値 c はおよそ 0.67 であることを示している。形状特徴値 \bar{C} は 0.665528 として抽出される。



図 2.8 凸包を用いた割合値の計算イメージ

この割合値 c は小さければ小さいほど角が尖ったトゲトゲとした花弁であり、大きければ大きいほど円形の丸い花弁であることが分かる。

今まで例で示してきた黄色で円形である花弁(図2.1など参照)の凸包を計算すると、 $\bar{C}: 0.982699$ となり、値からも円形の丸い花弁であることが判断できる。

2.5.3 距離特徴 \bar{D} の抽出

次に、花弁の距離特徴を抽出する。距離特徴と名付けているが、花弁の重心から各頂点までの距離の平均値を求めるため、形状特徴の一種だともいえる。特徴抽出のアルゴリズムを以下に示す。

- 1** 花弁領域の重心 G の計算を行い、画像内での座標 (g_x, g_y) を取得する。
- 2-1** 花弁の輪郭における各凹凸点 $R_i(r_{ix}, r_{iy})$ と $G(g_x, g_y)$ とのユークリッド距離 d_i を一周 360° 全て求める。
- 2-2** 求めた各ユークリッド距離 d_i に対して、最初に引かれた d_0 を基準線とし、各 $D_i = d_i/d_0$ を計算して正規化を行う。ちなみに $D_0 = d_0/d_0 = 1$ である。
- 3** 正規化が済んだ後は $D = \sum_{i=0}^n D_i$ よりユークリッド距離の平均値を出す。

以上の作業で計算された距離平均値 D を距離特徴 \bar{D} として抽出する。

実際にこのアルゴリズムで求められた距離特徴の値を図 2.9 に示す。

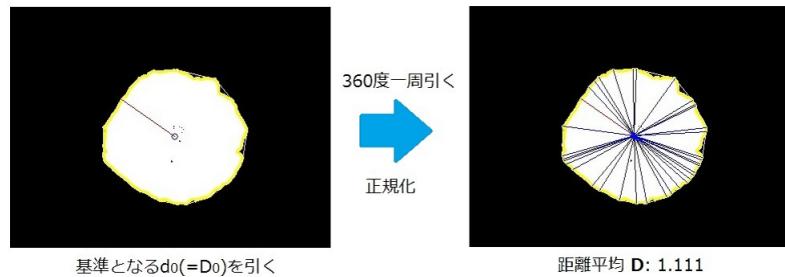


図 2.9 距離特徴を抽出するまでの流れ

この例では、1 本目に引かれた赤い線を基準 d_0 とし、重心 G から 360° 中の各凹凸点に対するユークリッド距離 d_i をそれぞれ求める。次に各 d_i を d_0 で割り正規化を行い、距離 D_i をそれぞれ算出した。正規化後に D_i のユークリッド距離平均値を求めると、 $\bar{D}: 1.111$ という結果になった。

2.5.4 特徴抽出のまとめ

以上 2.5.1 節から 2.5.3 節にかけて抽出した特徴の一覧を表 2.3 に示す。また、入力パターンの例として解説に用いてきた図 2.10 の各特徴値も同時に表 2.3 へと挙げておく。

図 2.10 参考入力パターン



表 2.3 抽出した特徴量データ一覧

特徴名	内容	参考入力パターンの各値
色特徴 \bar{H}	花弁領域内の色相の平均値	46
色特徴 \bar{S}	花弁領域内の彩度の平均値	97
色特徴 \bar{V}	花弁領域内の明度の平均値	79
形状特徴 \bar{C}	凸包領域内の花弁面積の割合値	0.983
距離特徴 \bar{D}	花弁の重心 G から各頂点までの平均距離値	1.111

計 5 特徴を特徴量データへと変換することができた。これら 5 つの特徴を基に、第 2 フェーズにて入力パターンの花の品種を識別する。

2.6 第3フェーズ 特徴量データを用いた品種の識別

第2フェーズでは、第1フェーズでカメラ映像内から抽出した各特徴量データを基に、機械学習 (*Machine Learning*) を用いて入力パターンの品種を識別・特定する。

入力パターンの各特徴量データは、5次元の未知パターンとして図2.11に示すような特徴空間へと写像される。図2.11の特徴空間は、クラスタの大きさによりクラスタ点の大きさを描画、似ている2クラスタを線分で繋ぐ、などの表現をすることにより高次元データを可視化したものである[1]。

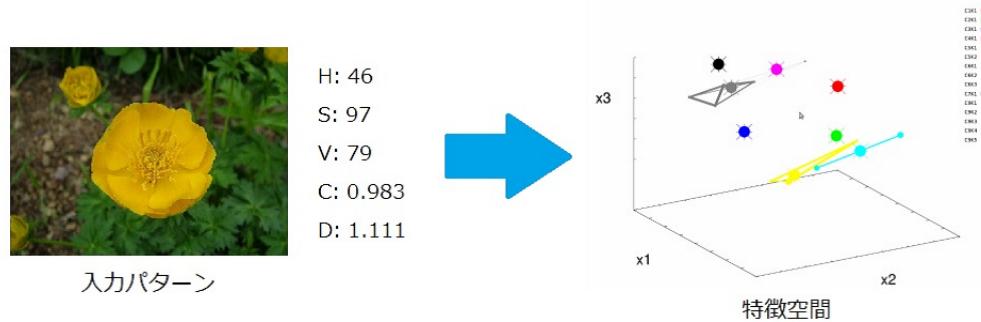


図2.11 入力パターンから特徴空間への写像

特徴空間内では、機械学習を用いて未知パターンと既存の学習パターン群との比較を行い、最も未知パターンに類似している学習パターンを見つけ出す。そして、その学習パターンの属するクラスを識別結果として出力する。類似した学習パターンを見つける役割を担う機械学習にも様々な種類があるが、本システムでは最も単純な手法である k 最近傍決定則 (k -nearest neighbor rule) を採用した。 k 最近傍決定則については次節で解説を行う。

2.6.1 k 最近傍決定則

k 最近傍決定則を解説する上で、最近傍決定則の理解は不可欠であるので先に説明しておく。

・最近傍決定則 (nearest neighbor rule) とは -

特徴空間上において、未知パターンから最もユークリッド距離で近接な学習パターンを見つけ出し、その学習パターンが属するクラスを識別結果として出力する。

最近傍決定則は、特徴空間上で近接しているパターン同士はその性質も互いに似ている[2]という概念に基づいていて、手法も単純明快で分かりやすい。しかし、もし最近傍にあったクラスが全く関係の無いクラスだった場合などを想定すると、誤識別をしてしまう可能性も否めない。

そこで、 k 最近傍決定則では、最近傍決定則を k 番目に近接な学習パターンにまで対して行い(但し k は奇数)、その k 番目までの識別結果の中で多数決を行って最終識別結果を決める。多数決を行うことにより、偶然混じっていた全く関係の無いクラスが最近傍に存在した場合でも、適切な識別結果が得られるという可能性が高い。

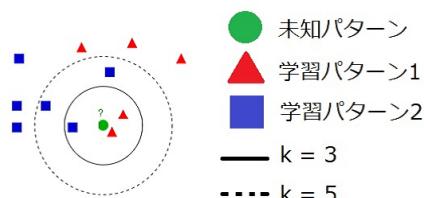


図 2.12 k 最近傍決定則のイメージ(仮)

図2.12を用いて k 最近傍決定則を解説すると、●(未知パターン)の識別結果は、 $k=1$ (最近傍)の場合は▲、 $k=3$ の場合も▲: 2, ■: 1より▲であるが、 $k=5$ にまで範囲を広げると▲: 2, ■: 3より識別結果が■に逆転するという現象が生じる。

結論として、 k 最近傍決定則は特徴空間によって適切な k の値を定めることにより、最近傍決定則より安定かつ正確な識別結果を期待できることが分かる。

2.6.2 学習パターン群の生成

次に、特徴空間内にて識別を行う際に、入力パターンとの比較をするために必要となる学習パターン群を生成する。

各学習パターンは2次元画像上で特徴抽出を行い、抽出アルゴリズムや特徴の種類、数などは、2.4節の第1フェーズで用いた手法と同じものを採用する。また、各クラスに対して、予め教師ラベルも付加しておくものとする。学習パターン群の例を図2.13に示す。



図2.13 教師ラベルを付加した学習パターン群

また、花の品種ごとにクラスを分類しても、それぞれパターンの数が少なければ学習パターンとして成立しないため、1クラスに対して最低10パターン以上を用意するものとする。1クラスの学習パターンの内訳も、単調で同じような画像ばかりではなく様々な角度や光加減から撮影された画像を用意する。例を図2.14に示す。



図2.14 各クラスの学習パターンの内訳

2.6.3 機械学習を用いた識別

前節までに準備してきた特徴量データや各学習パターンを基に、いよいよ入力パターンの識別を行う。識別を行う際にはOpenCVライブラリ内に予め用意されているMLアルゴリズム[3]から、 k 近傍法を使用した。

ここで品種特定までの流れをもう一度復習しておく。以下は2.3節で示したものと同じである。

第1フェーズ ウェブカメラから映像を入力し、閾値処理により花弁領域を抽出する。また、第2フェーズにおいて特徴抽出の精度を上げるためにノイズ除去やラベリング処理といった映像編集を行う。

第2フェーズ 第1フェーズで編集された映像に対して、花弁の平均色・形状などの各特徴を抽出。抽出された情報を特徴量データへと変換する。

第3フェーズ 第2フェーズで得られた特徴量データを基に、機械学習を用いて識別を行う。識別結果が出次第、品種を特定し結果の表示を行う。

以上の流れを1セットとし、このセットを延々と繰り返し続けることによって本論文の主題にも書かれているリアルタイム画像認識を実現できる。



図2.15 リアルタイム画像認識のイメージ

なお、本システムは3つのフェーズ全てをOpenCVで構築しているため、各フェーズをプログラム内で高速にループ処理し続けることによって映像のリアルタイム処理を可能としている。

2.7 システム稼動中の様子

実際にシステムを稼動させ、リアルタイム画像認識を実験している様子を図2.16に示す。

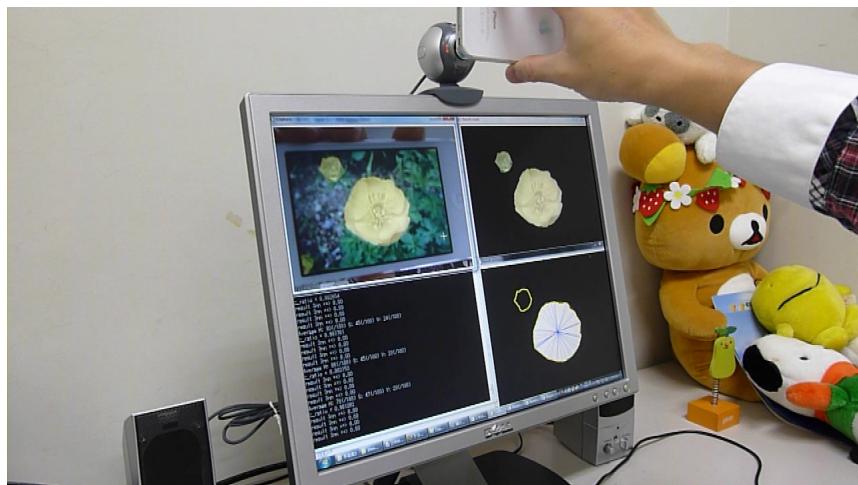


図2.16 システム稼動中の様子

PCモニタでは画面内を4分割しており、左上は今現在 webcam に映っているキャラチャ画像、右上は花弁領域抽出が行えているか確認のためのマスク画像、左下は識別結果表示をし続けるコマンドライン画面、右下は特徴抽出が行えているか確認のための2値画像をそれぞれ表示している。モニタ内の各画面から、この実験例では花弁の各情報を入力媒体から綺麗に抽出できていることが確認できる。

また、リアルタイム処理を行っているので、webcam の映像が変わる度に左下の識別結果もコマンドラインに更新されていく。本システムでは、識別を行う際におよそ 1 認識/0.5[sec] のスピードに設定している。

なお、この実験時の入力媒体は iPhone を用いた。もちろん、入力媒体はデジカメのディスプレいや、印刷された L 判写真を用いることも可能である。

第3章 評価実験

本章では、実際にシステムを稼動させ、評価実験を行った際の結果や考察などを述べる。

3.1 評価実験: 概要

システムの構築とテストが終了したところで、評価実験を行った。評価実験は複数の入力パターンに対して行うものとし、識別精度を数値で出すものとする。また、webcam にかざす入力媒体は iPhone のディスプレイを用いた。

実験の際に用意したクラスは代表的な 3 種類で、花弁が白色で丸い形状の花をクラス ω_1 、花弁が黄色で丸い形状の花をクラス ω_2 、花弁が白色で尖った形状の花をクラス ω_3 とした。3 クラスの各概要は、表 3.1 の通りである。

表 3.1 評価実験の対象クラスク一覧

			
クラス ω_1	クラス ω_2	クラス ω_3	
品種名	レブンアツモリソウ	レブンキンバイソウ	レブンウスユキソウ
花弁の色	白色	黄色	白色
花弁の形状	丸い	丸い	尖っている

これら各クラスは、以下のような類似点をもつ。

クラス ω_1 と クラス ω_2 花弁の形状が類似している(丸い)。

クラス ω_1 と クラス ω_3 花弁の色が類似している(白色)。

以上の類似点を正確に識別できるかがこの評価実験のポイントとなる。また、 クラス ω_2 と クラス ω_3 には類似点が無いため、 正確に識別を行える可能性が高いといえる。

3.2 評価実験: 結果

ここでは前節で述べた評価実験の結果を記す。 $k=3$ とした場合の識別結果を表 3.2 に示す。表 3.2 では行が入力パターンのクラス、列が入力パターンに対しての出力クラスを示している。表内の赤で示している数字は、それぞれ正しい識別に成功したことを表している。

ω_1 を例にとると、入力に対して ω_1 と出力されたのが 21 パターン中 15 パターン、 ω_2 と出力されたのが 0 パターン、 ω_3 と出力されたのが 6 パターンということである。よって、 ω_1 の誤識別は 21 パターン中 6 パターンであり、誤識別率はおおよそ 29% となった。

表 3.2 3 クラスの識別結果

IN \ OUT	ω_1	ω_2	ω_3	Error(%)
ω_1	15	0	6	6/21 ≈ 29%
ω_2	0	33	0	0/33 ≈ 0 %
ω_3	2	0	12	2/14 ≈ 14%

3.3 評価実験: 考察

評価実験の結果を基に、考察を行う。実験結果をまとめると以下のようになつた。

クラス ω_1 と クラス ω_2 (花弁色: 異なる、形状: 類似)

両品種の花弁の形状は類似しているものの、花弁色の識別によって識別率 100% の完璧な結果が得ることができた。

クラス ω_2 と クラス ω_3 (花弁色: 異なる、形状: 異なる)

両品種には類似している点が無いため、予想通り識別率 100% の完璧な結果を得ることができた。

クラス ω_1 と クラス ω_3 (花弁色: 類似、形状: 異なる)

両品種の類似点は花弁色で、異なる点は形状のみであったが、完璧な識別結果を得られることはできなかつた。クラス ω_1 を入力した際に ω_3 と出力された誤識別率は 29%、クラス ω_3 を入力した際に ω_1 と出力された誤識別率は 14% であり、正確な識別率はおおよそ 70~80% であることが分かつた。

以上のまとめから考察すると、花弁の形状に関わらず、花弁色が異なる花同士 (ω_1 と ω_2) の識別率は非常に高い (100%) のに対し、花弁の形状が異なり、花弁色が類似する花同士 (ω_1 と ω_3) の識別精度は高いといえども完璧ではない (70~80%) ということが分かつた。つまり、形状に関する特徴抽出がまだ不十分であることが評価実験によって判明した。

3.4 大規模クラス識別実験

次に、評価実験とは別に、システムが大規模クラス識別に対応できるかという実験を行った。識別に用いたクラスは評価実験に用いた3クラスと合わせて計8クラスとし、さまざまな花弁色や形状の花を用意した。



図 3.1 大規模クラス識別実験に用いた 8 クラス

図 3.1 の各クラスの識別結果を表 3.3 に示す。総計パターン数は計 161 となった。

表 3.3 8 クラスの識別結果

IN \ OUT	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	Error(%)
ω_1	15	0	6	0	0	0	0	0	29%
ω_2	0	24	0	0	9	0	0	0	27%
ω_3	0	0	6	8	0	0	0	0	57%
ω_4	2	0	4	22	0	0	0	0	21%
ω_5	0	6	0	0	17	0	0	0	26%
ω_6	0	0	0	0	1	12	0	0	8%
ω_7	0	0	0	0	0	0	13	2	13%
ω_8	0	0	0	0	0	0	0	10	0%

以上の結果から、大規模クラス識別にも対応可能と言って差し支えはなさそうだが、やはり花弁色が類似しているクラス同士での識別精度の低さが目立った。

第4章 今後の改善点と課題

本章では、当システムにおける今後の改善点と、残された課題について述べる。

4.1 現状における当システムの問題点

まず、当システムの現状での技術的な問題点を複数挙げ、それに対する改善策を述べていく。

類似色クラス間での識別精度の向上

本システムの入力パターンに対する特徴抽出数は5つだったが、実際に評価実験を行ってみると花弁の色特徴に頼った識別がほとんどで、類似色クラス同士での形状特徴、距離特徴を用いた識別は完璧とはいえない内容だった。そこで、類似色クラス間の識別率を更に上げるためにも、形状に関する特微量データ数を増加させなければいけないと考える。

花弁色選択のバリエーション

本システムでは、パターンを入力する際にまずユーザに花弁色を指定してもらい、その色をターゲット色として閾値処理を行う仕様にしているが、現状ではユーザが選択できる花弁色は6色しか無い。そこで、識別率を向上させるためにも、ターゲット色の閾値を更に細かく設定して選択できる花弁色のバリエーションを増加させたいと考えている。

機械学習をより近代的なものへ

本研究は、システムのリアルタイム動作を意識したために、識別を行う際に使用する機械学習は k 最近傍決定則を使用した。しかし、 k 最近傍決定則は手法が原始的で識別精度は高いとは言えないため、より識別精度が高く近代的なSVM(*support vector machine*)などを適用させたいと考えている。

更なる大規模クラス識別への対応

本システムは評価実験で3クラス、大規模クラス識別実験にて8クラスの花の識別を行ったが、現実には更に多数の花が存在し、北海道内だけでも約二千種が存在する[4]ため実用的だとは言い難い。そのため、識別可能となる花の品種数を増やしつつ、尚且つ識別精度の向上も見計らって学習パターン群のサンプル数も更に増加させなければいけないと考える。

4.2 今後増やしたい機能・装置

次に、今後システムを構築するにあたって増やしたい機能や装置などを挙げていく。

追加学習の導入

本システムでは、予め人の手によって学習をさせたパターンしか学習パターン群として扱えない。そこで、一度入力パターンとしてシステムに識別させた特微量データを、そのまま自動的に学習パターンへと応用できる追加学習の導入を検討する。この追加学習を行うことによって、システムは識別を行うたびに学習を繰り返して賢くなり、更なる識別率の向上を見込むことができる。

クラス分類の階層化

現時点の本システムでは、各特徴が最も似通っているクラスを機械学習によって識別し、結果を出すというアルゴリズムになっているが、識別を行う際に大分類→小分類→詳細分類→…といった階層化を行いたいと考えている。例として大分類に花弁の大まかな色、小分類に花弁の形状、詳細分類に花弁の詳細な色と分類したとすると、品種を識別する際に分類を追うことによって誤識別を減らせ、識別率の向上を見込むことができる。

花弁色の指定を不要に

現時点での本システムでは、システム開始時にユーザに花弁色を決定してもらい、それをターゲット色として閾値処理を行い花弁領域の分離抽出を行っていた。しかし今後は、ユーザの手間を減らすためにターゲット色を用いない分離抽出の手法を模索していきたい。花に関して言えば、画像は背景はたいてい緑色(草や葉など)なので、その点にも注目して考えたい。

高画質カメラの導入

本システムでは、webcamを使って映像の入力を行っていたが、今回使ったwebcamは市販で画質もあまり良いとは言えないものだったので、より専門的で高画質なカメラを用いて識別を行いたいと考えている。現在検討しているのは、ハンディカムなどのハイビジョンビデオカメラを撮影状態で入力に用いる、或いはwebcamにこだわるのであれば500万画素以上やオートフォーカス機能があるものを使用して実験を行いたい。

Androidへの移行

最近、Android向けのOpenCVが開発されてバージョンアップを繰り返しているため、Androidアプリケーションへの移行も検討している。当システムのプログラムをAndroidに書き換えて実装することになるが、スマートフォンなどのカメラはそのまま映像入力にも使えるため、高画質カメラを準備する必要もない。また、Androidアプリケーションは一般公開できるという利点があるため、利用者の増加も見込むことができる。

4.3 社会的貢献

本研究は「観光用」とあることから、社会貢献の一環として一般公開することを想定して開始した。観光用画像として識別対象物を花に決定し、花の島として有名であり豊富な固有種も存在する北海道宗谷地方の礼文島に協力を依頼し、許可を得ることができた。



図4.1 礼文島の位置

今後は、本システムをより実用化し、実際に現地の観光案内所などに当システムを搭載したパソコンを配置して、撮影した花の品種を知りたいという一般観光客に利用してもらいたいと考えている。

第5章 おわりに

本研究の最終目標は、識別精度を完璧にし、4.2節で述べたAndroidアプリのような形でも、4.3節で述べた特定地域での公開のような形でも、システムを実用化して世に送り出すことだと考えている。この最終目標へ向けて、進学先でも研究を続けて是非実現させたい。

また、もしシステムが実用化となった場合、観光地における地域活性化について画像認識技術の分野からどれだけ取り組むことができるか、などといった情報科学を用いた特定の問題解決にも非常に興味があるため、テーマが変わってもこの種の研究は今後しばらく取り組みたいと考えている。

第6章

謝辞

本研究を遂行し、ならびに本論文を作成するにあたって、釧路工業高等専門学校情報工学科 天元宏准教授に多くのご支援とご指導を頂きましたことを心より感謝いたします。

また、本研究の遂行にあたり協力してくださった北海道礼文島の皆様、その他の皆様にも御礼申し上げます。

参考文献

- [1] 高瀬陵平, 混合正規分布による高次元識別データの可視化.
–平成 23 年度 釧路工業高等専門学校情報工学科 卒業論文, 2012.
- [2] 石井健一郎, 上田修功, 前田英作, 村瀬洋, わかりやすいパターン認識.
–オーム社, 1998.
- [3] G. Bradski, A. Kaehler, *Learning OpenCV*.
–O'Reilly Media, 2008, 462–463.
- [4] 梅沢俊, 新北海道の花.
–北海道大学出版会, 2007.

発表論文・その他

レフリー無し論文

1. 児玉 拓巳, 観光用画像認識システムの構築.
創造性を育む「卒業研究」集, 国立高等専門学校機構, March 2012.