OpenCV を用いたりんごの等級判定

飯田聡美　(指導教員: 金子 晃)

1　はじめに

野菜や果物など生鮮食品には, その食品の優劣を決めるのに等級が使われている. 本研究では, りんごの等級判定を, 画像やデータを用い, パターン認識の方法で自動的に行うプログラムを考える. 画像解析には OpenCV ライブラリを利用する.
なお, 等級の決定基準は自治体により様々であるため, ここでは最も生産量の多い青森での基準を採用する. また, 品種は『サンふじ』を用いる.

りんごを比較するに当たっては
色・概形（傷の数）・重さ・直径・高さ・生産月
のデータを用いる

2　りんごの色の比較

2.1　比較の方法

画像の色を比較する手段として, HSV モデルを用い, ヒストグラムで比較する方法がある, RGB モデルでは色を赤 (Red)・緑 (Green),・青 (Blue) で表現したが, HSV モデルでは色相 (Hue)・彩度 (Saturation)・輝度 (Value・Brightness) で表現する.
また, ヒストグラムは通常だと横軸に階級, 縦軸に度数を取るが, 今回は横軸に色相, 縦軸に彩度とし, 輝度（度数）をグレイスケールで表現した３次元型のヒストグラムを用いる.

図 1: 特選

図 2: 特A

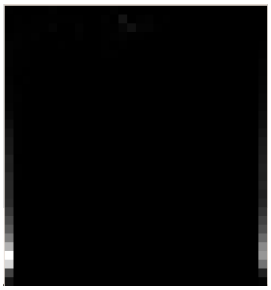


図 3: 特選

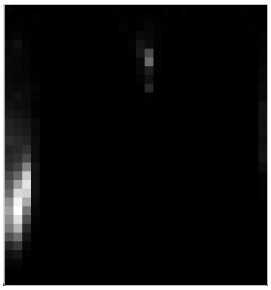


図 4: 特A

ヒストグラムで比較する場合, 次の４つの基準が使われる.

1. Correlation（相関）

$$d_{\mathrm{cor}}(H_1,H_2)=\frac{\sum_i H_1'(i)\cdot H_2'(i)}{\sqrt{\sum_i H_1'^2(i)}\cdot\sqrt{\sum_i H_2'^2(i)}}$$

$$\text{ただし } H_k'(i)=H_k(i)-\frac{1}{N}(\sum_j H_k(j))$$

2. Chi-square（χ２乗）

$$d_{\mathrm{chi}}(H_1,H_2)=\sum_i\frac{(H_1(i)-H_2(i))^2}{H_1(i)+H_2(i)}$$

3. Intersection（交差）

$$d_{\mathrm{int}}(H_1,H_2)=\sum_i\min(H_1(i),H_2(i))$$

4. Bhattacharyya distance（Bhattacharyya 距離）

$$d_{\mathrm{Bhat}}(H_1,H_2)=\sqrt{1-\sum_i\frac{\sqrt{H_1(i)\cdot H_2(i)}}{\sqrt{\sum_i H_1(i)\cdot\sum_i H_2(i)}}}$$

2.2　比較の結果

例として, 特選のりんごとその他のりんごを比較した結果を示す.

基準	Corr	Chi	Inter	Bhat
特選	0.938734	0.155357	0.789446	0.221927
特	0.744863	0.456616	0.594798	0.372669
特A	0.182469	1.344078	0.228811	0.729235
青秀	0.489605	1.046346	0.334630	0.642747

3　りんごの概形の比較

3.1　概形の抽出

りんごの概形を抽出するために以下の操作を行う.

1. RGB の色値を調整する (R を強調する)
画像中の各ピクセルに対して直接操作する.

2. RGB の画像を R,G,B の面に分割する
cvSplit 関数を使用する.

3. morphing を行う
morphology 演算として opening を使用する.

dilation（膨張）
注目するピクセル（原画像X中）が０の場合, マスク画像Yと重ねて１個でも１のピクセルがあれば注目するピクセルを１に変える.

erosion（収縮）
注目するピクセル（原画像X中）が１の場合, マスク画像Yと重ねて１個でも０のピクセルがあれば注目するピクセルを０に変える.

opening とは, まず erosion を行い, 次に dilation を行う演算のことである.

4. 得られた画像を２値化する
cvThreshold 関数を使用する.

5. 概形を探索する関数を用いる

3.2 概形の比較

概形の比較には Hu モーメントを使用する.
モーメントの式は

$$m_{p,q} = \sum_{i=1}^{n} I(x,y)x^p y^q$$

$I(x,y)$ はピクセル (x,y) の輝度, p,q はそれぞれ x 方向, y 方向の次数である.

$$\mu_{p,q} = \sum_{i=1}^{n} I(x,y)(x - x_{\text{avg}})^p (y - y_{\text{avg}})^q$$

が中心モーメントを表す式であり, また

$$x_{\text{avg}} = \frac{m_{10}}{m_{00}}, \quad y_{\text{avg}} = \frac{m_{01}}{m_{00}}$$

である. 更に中心モーメントに対して

$$\eta_{p,q} = \frac{\mu_{p,q}}{m_{00}^{(p+q)/2+1}}$$

が正規化された中心モーメントとなる. この中心モーメントの線型結合が Hu モーメントであり, l1 から l7 の 7 個がある.

4 機械学習

これまでの過程で得られたデータから, りんごの等級を判定するための機械学習を行う.

4.1 機械学習について

機械学習とは, データの集合から解析を行い, その中からルールやパターンを抽出して知識を得る手法のことである.
OpenCV は Machine Learning Library(MLL) を持ち, データの分類や回帰, クラスタリングに必要な関数とクラスがまとめられている.
以下で MLL に含まれている関数の一部を紹介し, りんごの等級判定のアルゴリズムとして実装してみる.

4.2 Decision tree

木構造を持つ分類器である.
データの中から, そのデータを最良に分割するような特徴や閾値を見つけ出す. それによりデータが分割され, 左右の枝のどちらかを進む. この手順を再帰的に繰り返す.
必ずしも最も効率の良い分類器とは言えないが, 処理が高速で高い機能を持つので, 最初に試すのに適している.

4.3 Boosting

分類器の集合を重み付けし, 結合することによって作られる.
まず, 弱い分類器を一つずつ学習させ, 同時にデータを正確に分類するようなそれぞれの分類器への重みも決定する. 重み付けを決定した後に再び学習させることで, エラーが生じた箇所を改良するように重みを更新する. この操作を, エラーが閾値以下になるまで繰り返す.
このアルゴリズムは, トレーニングデータが大量にある時に特に有効である.

4.4 Random tree

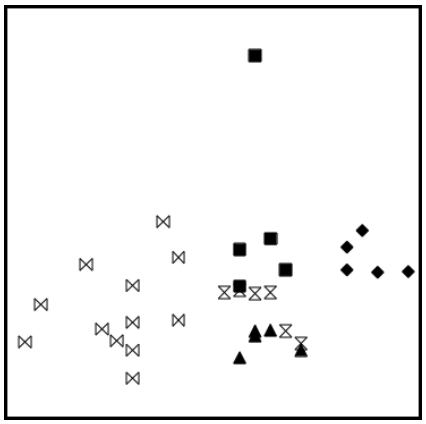
ある程度の深さまで分割されるような Decision tree の集合のこと.
学習の過程で, それぞれの木の各ノードにはランダムにデータの特徴が割り当てられ, それに従い分割する. これにより, それぞれの木は統計的に独立な決定因子となっていることが確かめられる. 木に対して重み付けはされない.
このアルゴリズムは大変有効であることが多く, それぞれの出力を木の数で割ることで, 回帰分析にもなる.

4.5 importance

Decision tree は, 各ノードでどの特徴が最適なデータの分割を行うかを調べる分類器である. 従って, 一番上のノードには最も重要な特徴が用いられ, 次の水準にあるノードは二番目に重要な特徴が用いられていることになる.
この重要度を OpenCV の関数では importance として定めている. Random tree では, Leobreiman が開発した手法により, 同様に importance が定められている.

4.6 実行結果

Decision tree を実行して得られた結果のうち, importance の大きい二つの特徴を二次元座標空間に図示したものが, 以下の図である.
実験には 5 種類の等級のりんごを使用しており, 同一のマークは同一の等級であることを示している.



5 まとめと今後の課題

本来機械学習には大量のサンプルが必要であるが, 今回は少量のサンプルであるにもかかわらず良い結果が出てしまったと言える. よって, サンプルに偏りがある可能性もある. 十分な量のサンプルでも同様の結果が得られるのか, 得られない場合はどのような特徴を追加することが適当なのか考えていくことが今後の課題である.

参考文献

[1] Gary Bradski,Adrian Kabler :
"Learning OpenCV", O'REILLY , 2008 .

[2] 石井健一郎, 上田修功, 前田英作, 村瀬洋 :
"パターン認識", オーム社 , 1998 .

[3] "OpenCV-1.1pre リファレンスマニュアル",
http://opencv.jp/opencv-1.1.0/document/