

J-15 OpenCV を利用した認識システムの学習と作成

発表者氏名 松橋 駿弥

担当教員 小笠原 祐治

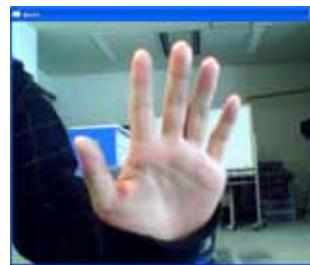
内かどうかを見て判断します。今回の肌色検出では、
YCrCb を用います。Cr が 133 ~ 173、Cb が 77 ~
177 の範囲内かどうか見ていています。

1 目的

コンピュータが人や物を認識する仕組みを理解する。
OpenCV を利用してコンピュータに任意の物体を
認識させ、処理を行うプログラムを作成することで
より理解を深める。

2 システム概要

カメラを通してグーチョキパーを認識し、PC とジ
ヤンケンをするジャンケンプログラムを作成する。

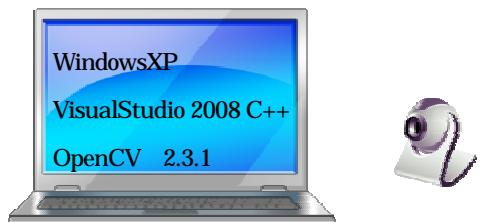


検出前

検出後



3 開発環境



4 プログラムの流れ

カメラから画像を取り込み以下の処理を行う
取り込んだ画像から肌色を検出。背景を白にする。
欠損領域を補間する。
分類器を使用して、手の形をグーチョキパーに分
類する
パソコン側の手をランダムに選択
ジャンケンの結果を表示

YCrCb とは

YCrCb は Y : 輝度 Cr : 色相 (青色の差分)

Cb : 色相 (赤色の差分)

RGB から YCrCb を求める式は以下の通り

R:赤 G:緑 B:青

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

$$Cr = 0.5 * R - 0.41869 * G - 0.0813 * B$$

$$Cb = -0.16874 * R - 0.33126 * G + 0.5 * B$$

6 欠損領域の補間

肌色検出した際に欠損した領域を補間する。1画
素ずつ見ていき、画素の左と上の画素が肌色の場合
は肌色に置き換える。



補間前

補間後

5 肌色検出について

色の検出方法は、色のパラメータが指定した範囲

7 手の形を分類する

取り込んだ画像を分類器 (OpenCV の関数、cvHa

arDetectObject)にかけ分類する。(分類器作成については 9 で説明する。)

例えば、グー用の分類器は画像にグーが含まれているかどうかを判別する。同様にチョキ、パーの分類器も用意する。

実時間で、入力画像を三つの分類器で 10 回判別し、最も検出数の多いものを分類結果とする。

グーの場合は 0、チョキの場合は 1、パーの場合は 2 を返す。

8 勝敗判定

グー：0 チョキ：1 パー：2 とする。

m を人、c をコンピュータとしたとき、

$(m - c + 3) \% 3$ が

0：引き分け 1：人の負け 2：人の勝ち
とする。

9 分類器作成

分類器を作成するために機械学習を行う。前回紹介した Haar-Like 特徴を用いた分類器を作成する。

機械学習には、サンプルデータとして、検出対象が写っている正解画像（グーチョキパーの三種類）と、写っていない不正解画像が必要である。

9.1 サンプルデータの作成

正解画像と不正解画像は任意の枚数だが、一般的に正解画像 7000 枚、不正解画像 3000 枚が推奨されている。不正解画像は一枚から切り取ることで複数の画像を生成できるので、実際は少なくともよい。

正解画像を作成するために、createsamples.exe を使用する。このソフトは、一枚の正解画像から回転や歪みをかけて複数枚の正解画像を作成することができる。グーチョキパーそれぞれの分類器が必要なため、それぞれに正解画像と不正解画像を用意する。

今回は、原画像を 20 枚ほど用意し、createsamples.exe にかけて、5000 枚ほどに増やした。

9.2 機械学習

機械学習には haartraining.exe を用いる。このソ

フトは、用意した正解画像と不正解画像から、物体を検出するためのパラメータを作成する。

実行にかかる時間は正解画像や不正解画像の中身や量に変わるが、今回の作成でかかった時間は以下の通り

正解画像 不正解画像 時間

2200 500 3 日

5000 3350 6 日

また、正解画像や不正解画像の中身によっては、機械学習が長時間かかり、中断した場合もあった。

9.3 検出例



正しくチョキと認識



誤ってチョキと認識

正解画像 2200 枚、不正解画像 1200 枚でチョキの分類器を作成した。

作成した分類器では、上の図のように、チョキを正しく認識しているが、パーも誤ってチョキと認識しまった。グー、パーの分類器も同様の結果だった。

正解画像の枚数や不正解画像の枚数を調整してみたが、同じ結果だった。

原因としては以下の事が考えられる。

- 不正解画像が適切でない
- 不正解の手の画像も必要？
- 手（肌色）の検出が正しく出来ない
- 欠損領域の補間が不適切
- 撮影環境の影響（明るさなど）

まとめ

図形処理工学の授業で図形処理の実習を行ったが、今回は、OpenCV を用いてよりレベルの高い認識処理に挑戦した。期待した結果とはならなかったが、新しい分野の勉強が出来たことが有意義だった。