

# Computer Vision Final Report

81819433 開放環境科学専攻情報工学専修 修士1年 飯塚 健介

2018年7月20日

## 1 目的と課題設定

私は主にセグメンテーションについて理解を深めるために授業で習った手法を用いてセグメンテーションの処理を自分で実装してOpenCVに備わっているライブラリ関数での実装と比較検討、また自分の実装について考察を行った。今回私がセグメンテーションを選んだ理由は授業中に聞いた手法でなぜセグメンテーションができるのかが気になったからである。また研究として深層学習のアクセラレータの開発を行うために深層学習について調べていたときにセグメンテーションは自動運転や医療画像の解析などにも利用されることを知ったので応用技術としても今後ますます注目されるのだろうと考えた。

## 2 開発及び実行環境

開発及び実行環境は次のとおりである。OpenCVは比較と入出力を受け取るために利用した。

- OS : Ubuntu18.04
- 使用言語 : Python3.6.5, C++11
- 使用ライブラリ : Numpy, Matplotlib, OpenCV

## 3 k-means 法を用いたセグメンテーション

まず、k-means 法を用いたセグメンテーションについて検討と実装を行った。

### 3.1 k-means 法について

アルゴリズムについて検討した。k-means ではいくつのクラスタに分類するかを人が指定しなければいけない。そしてランダムにクラスタリングされた 3 次元のベクトル（画像の RGB をベクトルの各要素とした）について各クラスタの重心を求める。その重心から各ベクトルのユークリッド距離を算出し最小となるクラスタに属するようとする。これをクラスタリング結果が前回のクラスタリング結果と一致するまで、もしくはある上限回数を行い、各クラスタの重心を決定する。ここに新たな入力ベクトルが入ってきた場合には各クラスタの重心とのユークリッド距離を算出し最小となったクラスタに属するようとする。具体例として、ランダムに生成した 2 次元ベクトルを基にしてクラスタの重心（オレンジの四角形でプロット）を求めて、入力データがどのクラスタに属しているかを図に示す。バツマークがランダムに生成したベクトル、丸マークで示しているのが入力データである。

ユークリッド距離が一番近いクラスタに分類されているのがわかる。

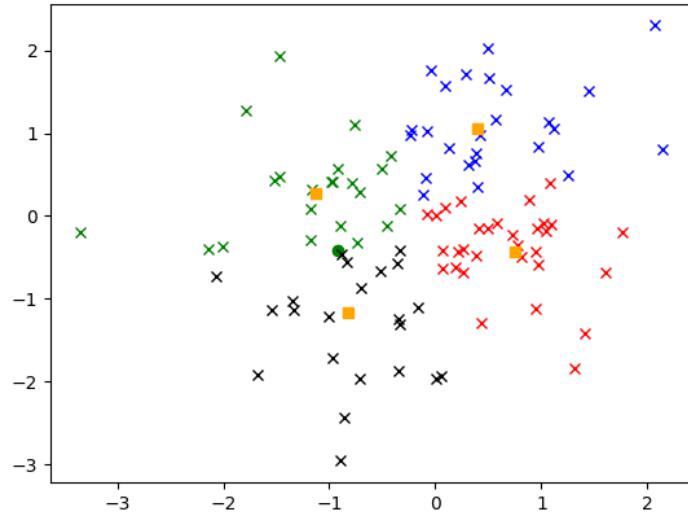


図 1 k-means による 2 次元ベクトルのクラスタリングの様子

### 3.2 実装方法

画像の入出力のみに OpenCV を使い、他の処理については numpy の行列、ベクトル演算を自分で実装した。

### 3.3 結果と考察

以下に何枚かの画像について、自分で実装した k-means での処理結果、対象画像、OpenCV の k-means の関数 (cv2.kmeans) による処理結果を示す。クラスタ数はすべて 8 で統一した。また自分の実装、OpenCV の kmeans とともに、各クラスタの色の塗り方はクラスタ重心となる RGB 値を用いた。



図 2 隠岐の島での豪華ディナーのオリジナル画像



図3 隠岐の島での豪華ディナー (自作 k-means)



図4 隠岐の島での豪華ディナー (cv2.kmeans)

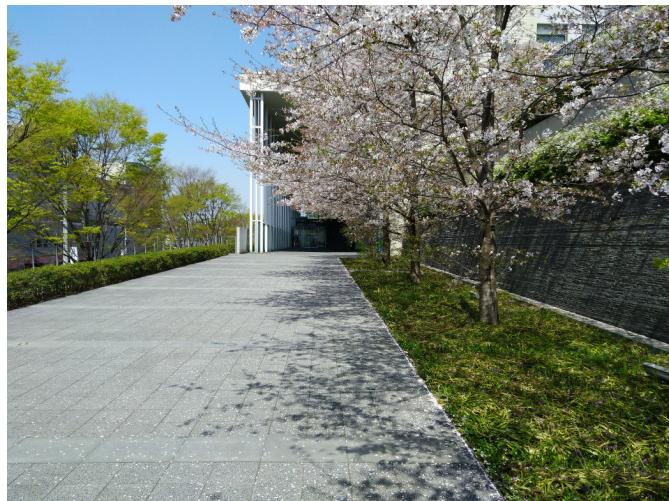


図5 卒業式の日吉キャンパスのオリジナル画像



図8 厄除けの川崎大師オリジナル画像



図 6 卒業式の日吉キャンパス (自作 k-means)



図 7 卒業式の日吉キャンパス (cv2.kmeans)



図 9 厄除けの川崎大師 (自作 k-means)



図 10 厄除けの川崎大師 (cv2.kmeans)

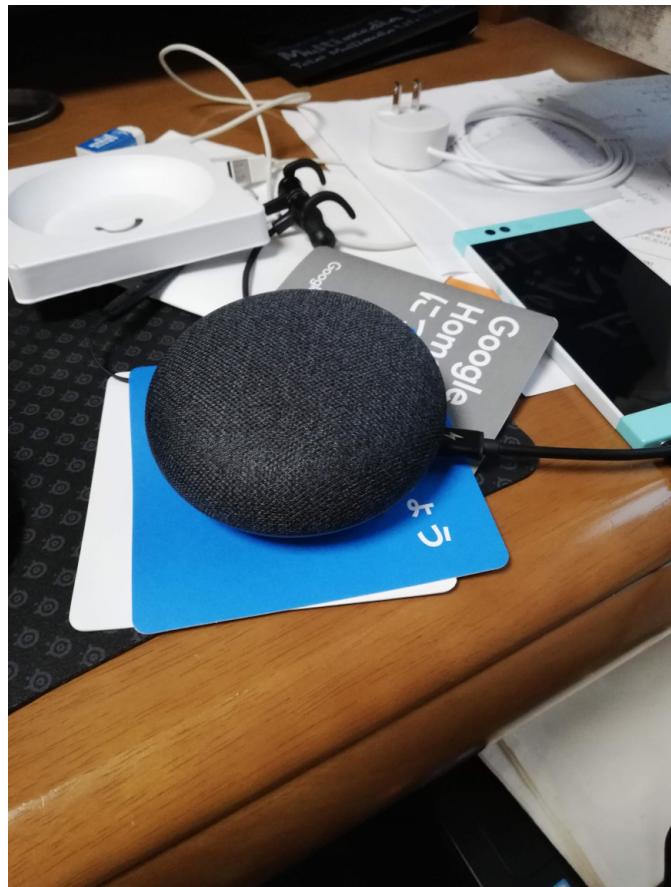


図 11 自宅の作業机のオリジナル画像



図 14 都庁展望台からの眺めのオリジナル画像



図 12 自宅の作業机 (自作 k-means)

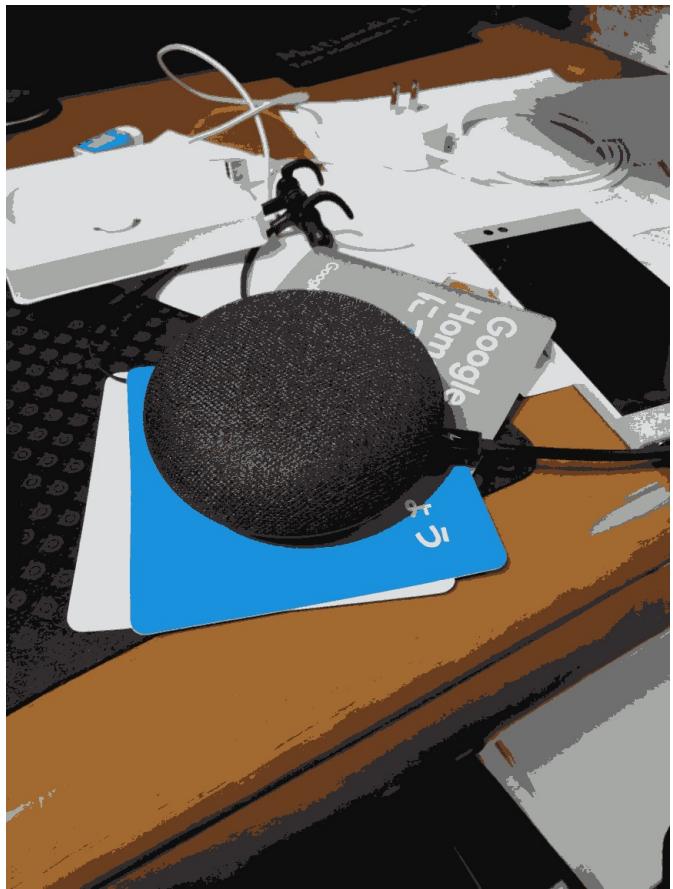


図 13 自宅の作業机 (cv2.kmeans)

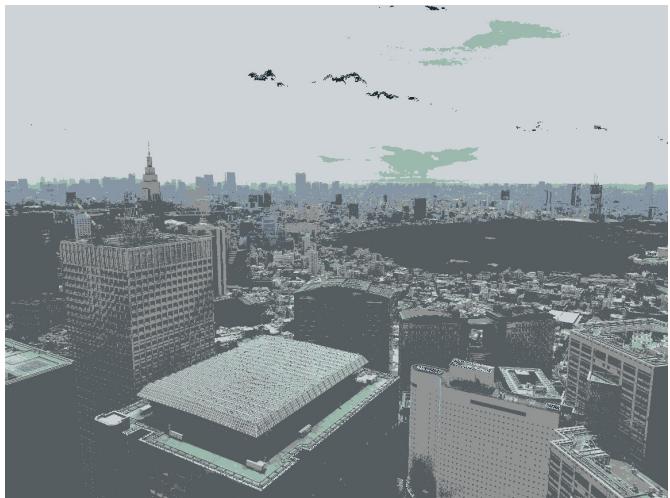


図 15 都庁展望台からの眺め (自作 k-means)



図 16 都庁展望台からの眺め (cv2.kmeans)

## 4 混合ガウシアンモデルを用いたセグメンテーション

### 4.1 混合ガウシアンモデルについて

混合ガウシアンモデルとは確率分布としてよく用いられるガウス分布の確率分布関数を複数重ね合わせたものである。

具体例として、k-means 同様に、ランダムに生成した 2 次元ベクトルを基にしてクラスタの重心を求めて、入力データがどのクラスタに属しているかを図に示す。バツマークがランダムに生成したベクトル、丸マークで示しているのが入力データである。ユークリッド距離が一番近いクラスタに分類されているのがわかる。

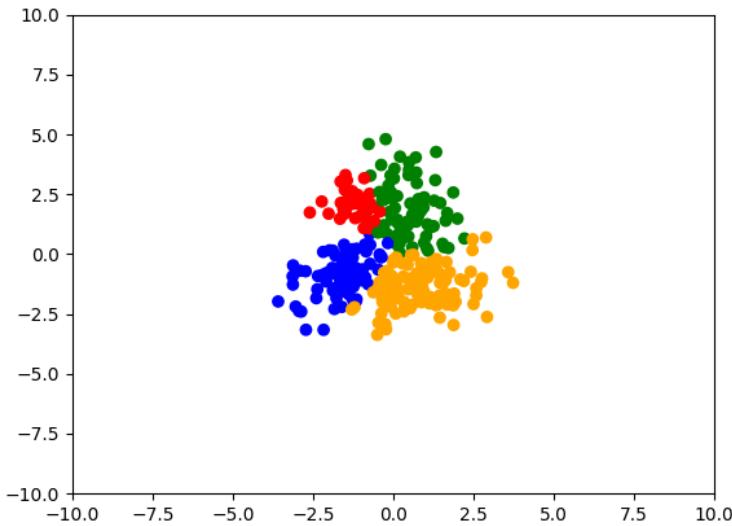


図 17 混合ガウシアンモデルによる 2 次元ベクトルのクラスタリングの様子

### 4.2 実装方法

上図の 2 次元でのクラスタリングのために Numpy で実装したコードのベクトルを 3 次元に拡張させて画像のセグメンテーションを行おうとしたが、上手く実装できなかった。色々と試行錯誤をしたのでソースコードの一部を以下に示す。

```
import matplotlib.pyplot as plt
import numpy as np

class GaussianMixture(object):

    def __init__(self, n_component):
        # Number of gaussian distribution
        self.n_component = n_component

    # 最尤推定
    def estimation(self, X, iter_max=100):
```