

## コンピュータビジョン自由課題説明書

### 0. 完成画像



図 1: 完成画像(final\_output.jpg)

### 1. 入力画像





girl2.jpg



girl3.jpg



girl4.jpg

## 2. 使用したクラス

- 既存のもの
  - Chromakey.java
  - JpegFileReader.java
  - JpegFileWriter.java
  - KMeans.java
  - MyImage.java
  - Tiling.java
  - Rotation.java
- 改変または自作
  - CvMain.java : 本体
  - GammaCorrection.java : 引数で指定した値から新しい RGB 値を得る
  - Scale.java : 引数で指定した大きさに拡大縮小
  - VirtualStudioRandom.java : ランダムな位置に画像合成 (クロマキーあり)
  - VirtualStudioSelect.java : 引数で指定した位置に画像の合成 (クロマキーあり)
  - VirtualStudioSimple.java : 引数で指定した位置に画像の合成 (クロマキーなし)

### 3. 手順



(左から output1.jpg, output2.jpg, output3.jpg)

(※output3.jpg が白枠でわかりづらいため、線をつけている。)

<output1.jpg>

① 最背画像の作成(imageProcessing2()にて作成)

ワインレッド色の画像(1703px\*1600px)を用意する。

入力画像のうち、pic0.jpg~pic10.jpg をクロマキー処理する。

クロマキー処理後の画像を、VirtualStudioRandom クラスを用いてワインレッド色の画像に合成していく。

この際 Rotation クラス、Scale クラスを用いることで、ループを回すごとに画像の大きさ、位置、角度の異なる画像が合成されるようにした。

このことにより、一番上に来る画像がしっかりと見える、画像一面がフルーツや植物で敷き詰められているような画像を作成できた。

copy.jpg ができる流れ。



ループ開始直後



ループ中盤



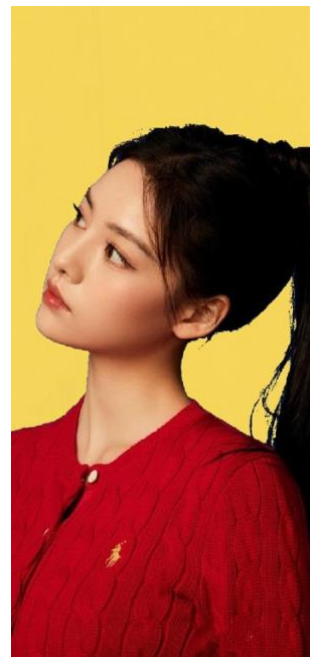
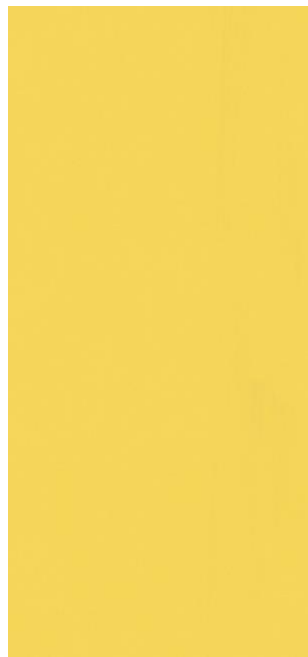
完成！

(毎回乱数で位置が決まるため、提出した画像とは別の画像が出来上がっている。)

② girl4.jpg, yellow.jpg を合成(imageProcessing3()にて作成)

girl4.jpg をクロマキー処理、拡大処理後 VirtualStudioSelect クラスを使って girl4.jpg, yellow.jpg を合成。

この際、引数で2枚の画像の位置関係を指定できるようにすることで、ちょうど良い位置で合成した。



(左から girl4.jpg, yellow.jpg, yellow\_girl4.jpg)

- ③ copy.jpg, yellow\_girl4.jpg を合成(imageProcessing4())にて作成)  
VirtualStudioSimple クラスを使って copy.jpg, yellow\_girl4.jpg を合成。  
この際、引数で 2 枚の画像の位置関係を指定できるようにすることで、  
yellow\_girl4.jpg がちょうど copy.jpg の真ん中に来るように合成した。



(左から yellow\_girl4.jpg, copy.jpg, output1.jpg)

<output2.jpg>

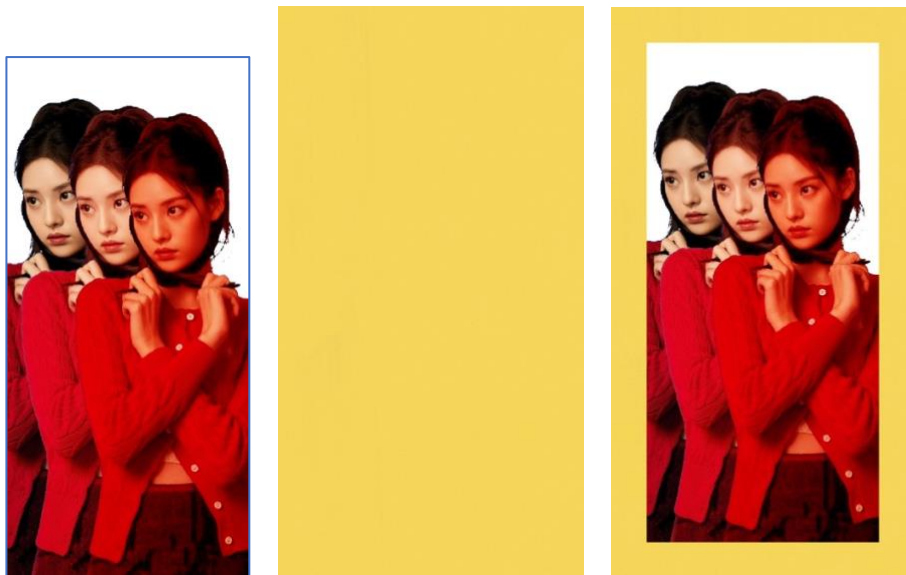
- ① gum2\_white\_girl3.jpg を作成(imageProcessing3())にて作成)  
GammaCorrection クラスを用いて処理後、VirtualStudioSelect クラスを用いて 2  
回合成。この際、徐々に画像が赤くなるように、右下に合成されるよう数値を調整  
した。





(※gum2\_white\_girl3.jpg が白背景でわかりづらいため、線をつけている。)

- ② gum2\_white\_girl3.jpg, bigyellow.jpg を合成(imageProcessing4())にて作成  
 VirtualStudioSimple クラスを使って gum2\_white\_girl3.jpg, bigyellow.jpg を合成。  
 この際、引数で 2 枚の画像の位置関係を指定できるようにすることで、  
 gum2\_white\_girl3.jpg がちょうど bigyellow.jpg の真ん中に来るように合成した。



(左から gum2\_white\_girl3.jpg, bigyellow.jpg, output2.jpg)

(※gum2\_white\_girl3.jpg が白背景でわかりづらいため、線をつけている)。

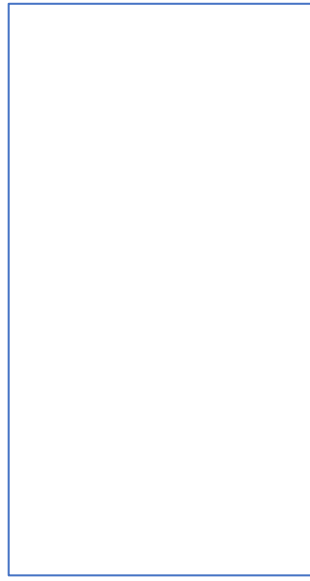
<output3.jpg>

- ① copy.jpg, girl2.jpg を合成(imageProcessing3())にて作成  
girl2.jpg をクロマキー処理、拡大処理後 VirtualStudioSelect クラスを使って girl2.jpg, copy.jpg を合成。この際、引数で2枚の画像の位置関係を指定できるようにすることで、ちょうど良い位置で合成した。



(左から copy.jpg, girl2.jpg, Flower\_girl2.jpg)

- ② Flower\_girl2.jpg, bigWhite.jpg を合成(imageProcessing4())にて作成  
VirtualStudioSimple クラスを使って Flower\_girl2.jpg, bigWhite.jpg を合成。  
この際、引数で2枚の画像の位置関係を指定できるようにすることで、  
Flower\_girl2.jpg がちょうど bigWhite.jpg の真ん中に来るように合成した。



(左から Flower\_girl2.jpg, bigWhite.jpg, output3.jpg)

(※bigWhite.jp, output3.jpg が白背景でわかりづらいため、線をつけている。)



<最後に>以下の output1.jpg,output2.jpg,output3.jpg を横一列に並べる。  
Tiling クラスを用いて並べる。



(※output3.jpg が白背景でわかりづらいため、線をつけている。)



～完成～

#### 4. こたわった点

copy.jpg を 11 個の画像ファイルから作ったところが一番大変であり、やりがいがあった。fruits ファイル内の 10 個の画像をクロマキー処理するのに一つ一つちょうどいい数値を探すことがとても大変だった。また、合成する時には位置を乱数で決めたり、たまに縮小したり、回転させることによってフルーツや植物が敷き詰められている画像を作成した。

全体の配色バランスにもとても拘り、女性の服、壁紙を始めとしてフルーツや植物の色にも統一感を持たせるため、時間をかけて画像を集めた。