コンピュータビジョン説明書

使用クラス

既存のもの

・JpegFileReader.java
・JpegFileWriter.java
・Chromakey.java
・Kmeans.java
・AlphaBlending.java　　：アルファの定数の値だけ変更した
・GammaCorrection.java　：定数の値だけ変更した
・Scale.java　　：SCALEX と SCALEY の定数の値だけ変更した

改変

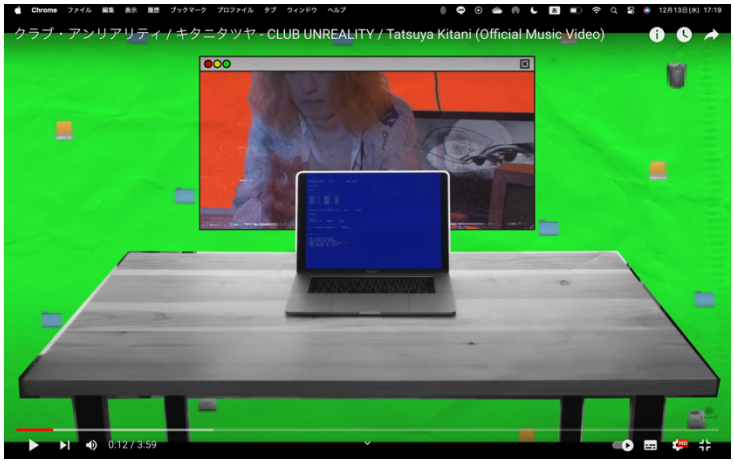
・SpaceFiltering.java　　：引数に double filter[]を追加し、フィルターを指定できるようにした
・Binalization.java　　：引数に binalization の閾値を追加した
・CvMain.java　　：メインクラス

自作

・Mosaic.java　　：モザイクをかける。粒度を指定する。引数を 1 つ増やして、モザイクをかける範囲を指定することも可能。
・AddGreenback.java　　：閾値を 0 にすると背景透過の画像にグリーンバックをつけ、rgb 合計値の閾値を引数におくと、その閾値を超えた箇所にグリーンバックをつける。（このクラスのメイン関数を使って処理した画像は既に desk.jpg としてフォルダ内にある。先に実行しないと、メインのコードがうまく動作しないため。）
・GetGreenIndex.java　　：緑の長方形の左上の座標と右下の座標を返す。
・GetGreenIndexTest.java：GetGreenIndex.java が動いているか確認するテスト。GetGreenIndex.java の返り値を Mosaic.java に入れて動かなかったときに、Mosaic.java のバグなのか、GetGreenIndex.java のバグか判定するために入れた。
・ClubUnreality.java　　：window 画像以外の画像を組み合わせる。
・AddWindow.java　　：window 画像を追加する。
・Posterize.java　　：ポスタリゼーション効果を足す。

完成画像（copy. jpg）

キタニタツヤの「クラブ・アンリアリティ」の以下のシーンを再現してみたくなったので、自分風にポップに再現してみた。

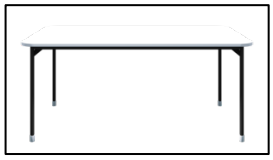


https://youtu.be/Ps2wPVtNZ3M?si=6g7QoRJYk1JMWsI3&t=12

入力画像

desk.jpg

 desk.jpg

 (←元画像 desk.png)

AddGreenback.java というコードを新規作成し、実行した。(コードは提出フォルダ内に含めた) これにより元々背景透過だった desk.png が緑の背景が足された desk.png に変換された。

https://commons.nicovideo.jp/works/agreement/nc316039, ” テーブル 2 のライセンス - ニコニ・コモンズ”, (参照 2023-11-23)

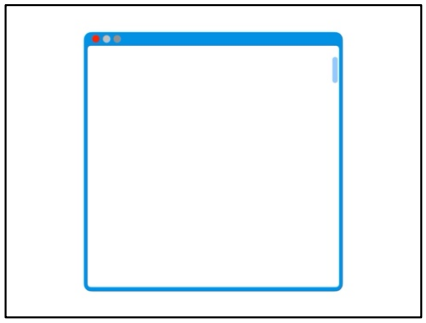
macbook.jpeg

https://www.photo-ac.com/main/detail/26425922&title=%E3%83%8E%E3%83%BC%E3%83%88%E3%83%91%E3%82%BD%E3%82%B3%E3%83%B3, “ノートパソコン - No: 26425922｜写真素材なら「写真 AC」無料（フリー）ダウンロード OK “, 写真 AC , (参照 2023-11-23)

folder.jpeg

自身の macbook の画面をスクリーンショットしました。

window.jpeg



https://www.ac-illust.com/main/detail.php?id=24563561&word=PC%E3%82%A6%E3%82%A3%E3%83%B3%E3%83%89%E3%82%A6%E6%AD%A3%E6%96%B9%E5%BD%A2%E3%83%95%E3%83%AC%E3%83%BC%E3%83%A0A%EF%BC%9A%E9%9D%92, “PC ウィンドウ正方形フレーム A：青イラスト - No: 24563561｜無料イラスト・フリー素材なら「イラスト AC」”, イラスト AC, (参照 2023-11-23)

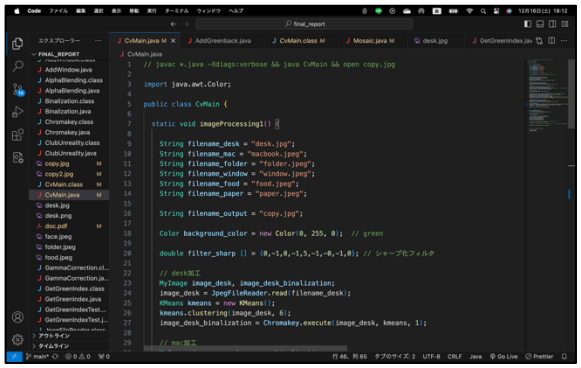
food.jpeg

「恋愛酒場メイ子」のお料理を自身で撮影した写真に、MacBook の Folder に備わっている機能で、緑の長方形を追加した画像。

paper.jpeg

https://www.photo-ac.com/main/detail/27905165&title=%E6%89%8B%E6%8F%89%E3%81%BF%E5%92%8C%E7%B4%99%E3%81%AE%E3%83%86%E3%82%AF%E3%82%B9%E3%83%81%E3%83%A3%EF%BC%88%E3%83%9B%E3%83%AF%E3%82%A4%E3%83%88%EF%BC%89, “手揉み和紙のテクスチャ（ホワイト） - No: 27905165｜写真素材なら「写真 AC」無料（フリー）ダウンロード OK”, 無料写真素材「写真 AC」, （参照 2023-12-07）

screen.jpeg



```
// javac *.java -Xdiags:verbose && java CvMain && open copy.jpg

import java.awt.Color;

public class CvMain {

  static void imageProcessing1() {

    String filename_desk = "desk.jpg";
    String filename_mac = "macbook.jpeg";
    String filename_folder = "folder.jpeg";
    String filename_window = "window.jpeg";
    String filename_food = "food.jpeg";
    String filename_paper = "paper.jpeg";

    String filename_output = "copy.jpg";

    Color background_color = new Color(0, 255, 0);  // green

    double filter_sharp [] = {0,-1,0,-1,5,-1,0,-1,0}; // シャープ化フィルタ

    // desk加工
    MyImage image_desk, image_desk_binalization;
    image_desk = JpegFileReader.read(filename_desk);
    KMeans kmeans = new KMeans();
    kmeans.clustering(image_desk, 6);
    image_desk_binalization = Chromakey.execute(image_desk, kmeans, 1);

    // mac加工
```

VSCode の画面をスクリーンショットした。

画像処理の手順

desk

Kmeans と Chromakey を用いて、2 値化を行った。

・Kmeans の出力結果

clustering: counter=1 distance=72030.0

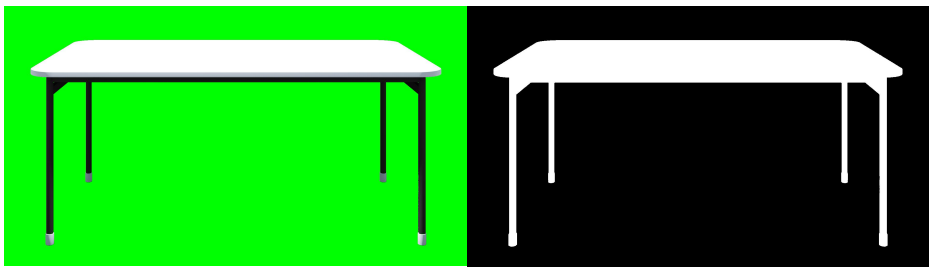
clustering: counter=2 distance=13177.0

clustering: counter=3 distance=67.0

clustering: counter=4 distance=15.0

clustering: counter=5 distance=4.0

・Chromakey.execute の引数を(image_desk, kmeans, 1)として、1 にすることで緑だけを切り取った。



（左：desk.jpg, 右：2 値化した結果）

Desk の元画像と 2 値化画像を ClubUnreality に入れて、メイン画像に追加する。

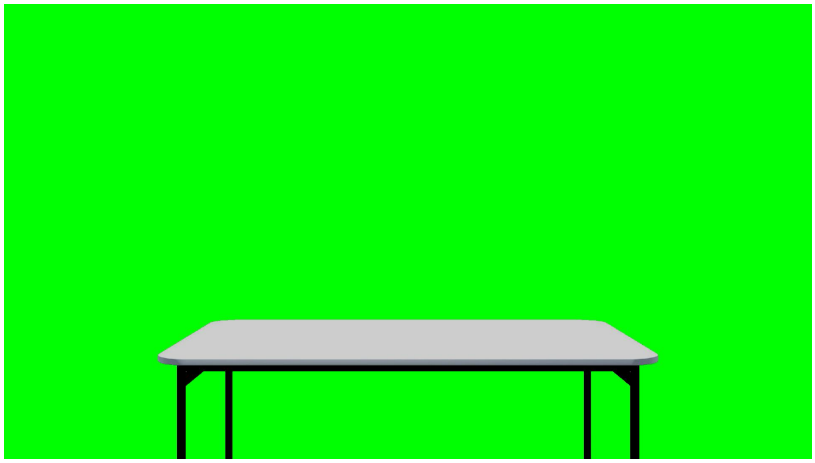
ClubUnreality 内では、

・2 値化したものを用いて、背景透過を作成した。（手順としては、2 値化画像の getRed() の値が存在するかを確認した。）

・desk を暗く加工する。（そのままだと明るすぎて他と馴染まないと感じたため）

・サイズと位置を調整。

ここまでのメイン画像



Mac

Binalization を用いて、2 値化する。(2 値化したものを用いて、背景透過画像を作成するため)

(左：mac.jpeg, 右：2 値化した結果)

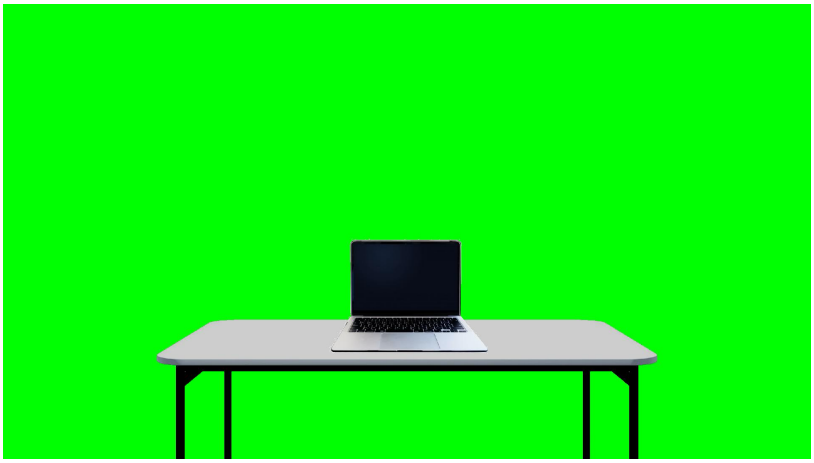
Mac の元画像と 2 値化画像を ClubUnreality に入れて、メイン画像に追加する。

ClubUnreality 内では、

・2 値化したものを用いて、背景透過を作成した。(手順としては、2 値化画像の getRed() の値が存在するかを確認した。)

・サイズと位置を調整。desk の位置から計算して、位置を設定。

ここまでのメイン画像



Folder

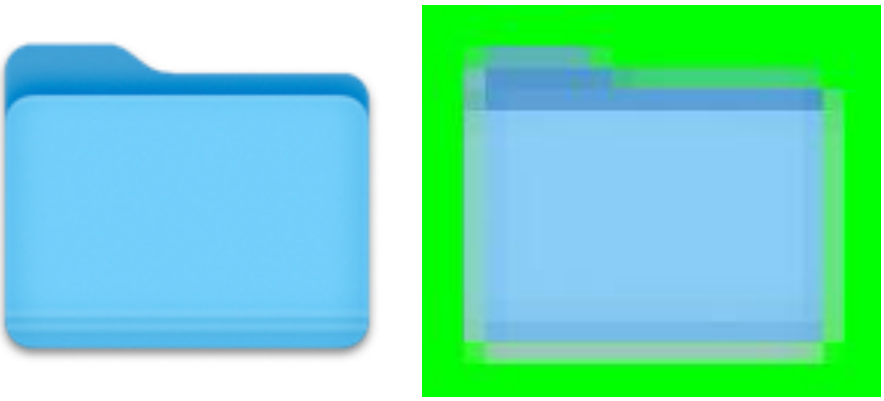
Mosaic クラスを用いて、モザイク加工を行った。

・複数の画素の平均値を出力画素とする。

・モザイクの粒度が画像の縦と横の画素値の約数ではない場合、指定した粒度でうまくいかなくなるので、int counter と例外処理を用いて、途中までの画素で計算できるようにした。

・モザイクの粒度は入力画像と相関関係を持たせた数値にしたことで小さい画像に対しても粒度が大きすぎないようになっている。

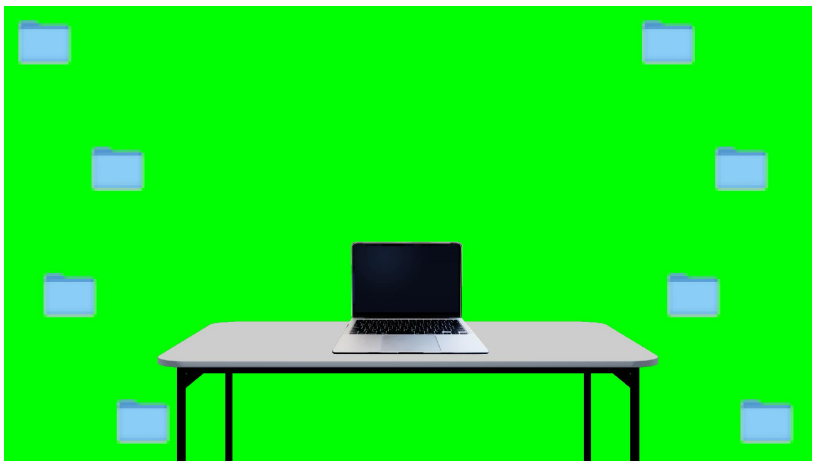
・(モザイクを部分的にかけたいときは引数に範囲を指定できるが、今回は指定していない。詳しくは food の加工で。)



（左：folder.jpeg，右：モザイク加工した画像）

ClubUnreality 内では、フォルダーを複製して、複数の位置を設定している。desk と重ならないよう、desk のサイズなどを考慮して位置を設定。

ここまでのメイン画像（MyImage Image_ClubUnreality）



ここから AddWindow クラスで追加する画像の加工手順に移る。

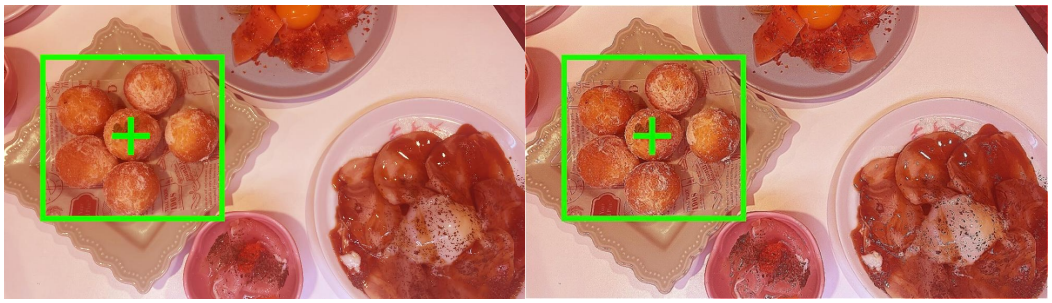
Food

Scale で縦横ともに 0.5 倍した。

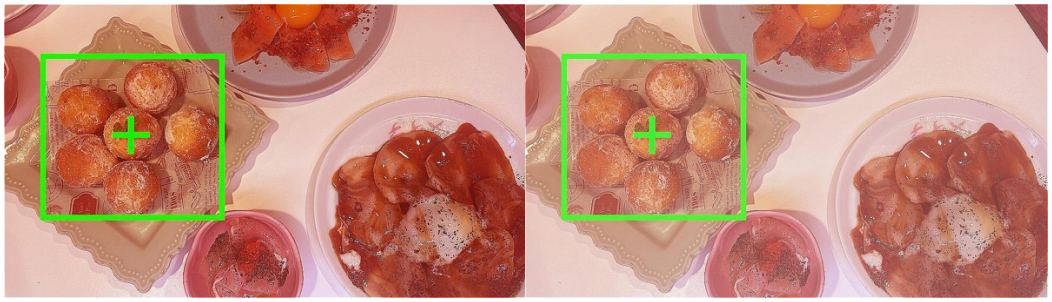
ただ美味しい食べ物を載せるのではこの世界観に合わないと思い、色味などの加工を行った。

・GammaCorrection で赤の要素を強くした。

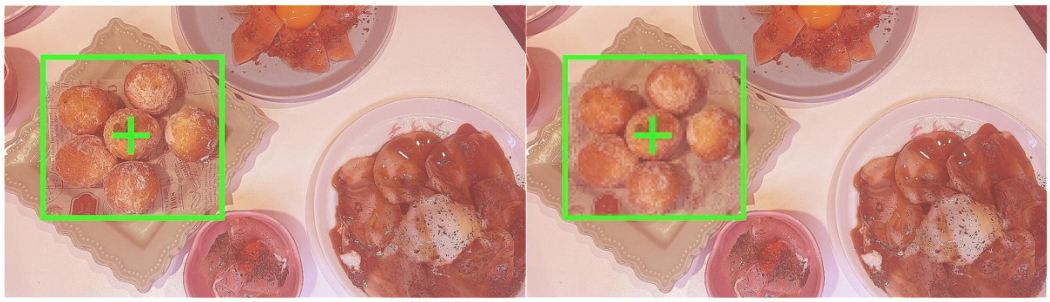
・SpaceFiltering でシャープ化フィルタをかけた。



・AlphaBlending で紙の画像をアルファブレンディングして加工（全体と合わせた時に強すぎたので）



・Mosaic で緑の枠内にモザイクをかけた。
GetGreenIndex で、緑の長方形の左上の座標と右下の座標を返す。返された GreenIndex をもとに、緑の長方形の内部に Mosaic.java でモザイクをかける。folder でも利用した Mosaic.java に、index の値を引数で加えても動くように書き換えた。

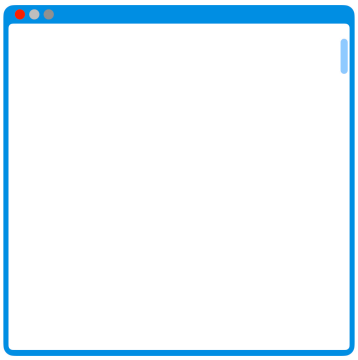


（Index の取得が正しいかどうかは、GetGreenIndexTest で確認できる。正しく取得できていれば、長方形の内部が黒くなる。）

AddWindow 内で、位置を調整して、かつ、Image_ClubUnreality が緑のところ（背景であるところ）だけに画素を置くようにした。（これにより mac の画像の手前に画像が来るのを阻止した。）

Window

SpaceFiltering でシャープフィルタをかけた。スライダーなどを見るとわずかだがシャープになった。



（左：window.jpeg，右：シャープ化した画像）

その際 mac の後ろ側に来るようにここで調整を行なった。AddWindow 内で、背景透過と mac の画像の奥に置くように調整する。

・Image_ClubUnreality が緑のところ（背景）であり、かつ、Window が白でない部分だけ、画素を置く。

ここまでの画像（Window, food を追加した）

ポスタリゼーション

・Posterize を用いて、ポスタリゼーションの効果を足す。

R, G, B のそれぞれの値を 10 の位で四捨五入したものを用いて、色の数を減らすことで、ゲームのような世界観に仕上げた。モザイク×ポスタリゼーションはゲームらしさを演出するのに、相性が良いことがわかった。

今回こだわったポイント

・Git で管理を行ったこと。

〇どのファイルをいじったのかわかったので、レポートが書きやすくなった。

〇途中までコードを書いているときの出力画像が git 上に保存されていたため、過去画像の比較がしやすく、どの加工がいいか吟味することができた。

×区切りの良いタイミングで commit をするのを忘れていることが多かった。付け足した機能ごとなどで今後は commit をする癖をつけていきたい。

・同じような結果になる処理を別の処理方法で試した。

〇Kmeans&Chromakey、Binalization と 2 通りでクロマキーを実現したり、SpaceFiltering、AlphaBlending の 2 通りでフィルタ加工を実現したりした。

・デバッグのための Test 関数を作ったこと。