

3年15組92番 全 海成(JEON HAESUNG)

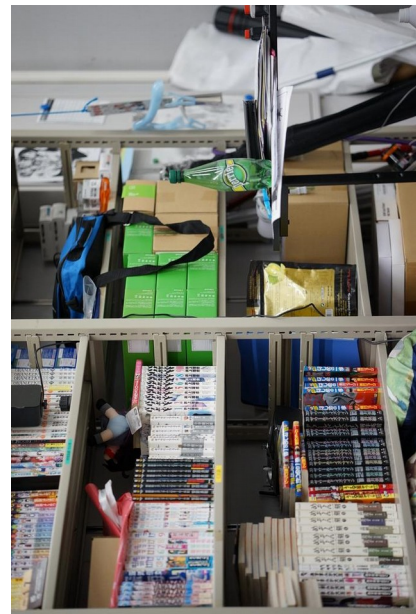
コンピュータサイエンス実習 A マルチメディア 一回目 レポート

必須課題 1

この課題のプログラムは JPG ファイルを入力としてとって横軸と縦軸を変えた JPG ファイルを出力に出すプログラムです。使用例は以下のようです。

```
./week1_ass1 [Input Image] [Output Image]
```

左のイメージを入力でプログラムを実行した結果が右のイメージです。

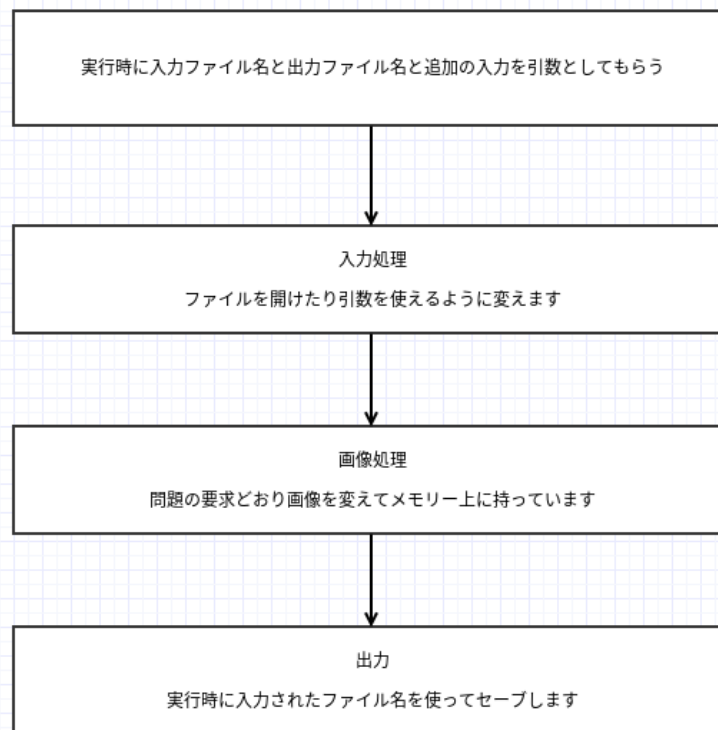


ワードの都合で図で正確にサイズが変わったのを見せるのが難しいのでターミナルで実行した結果です。

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ls
src.JPG week1_add1.c week1_add2.c week1_ass1.c week1_ass2.c week1_ass3.c week1_ass4.c week1_ass5.c
week1_add1 week1_add2 week1_ass1 week1_ass2 week1_ass3 week1_ass4 week1_ass5
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ./week1_ass1 src.JPG dst.JPG
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ls
dst.JPG week1_add1 week1_add2 week1_ass1 week1_ass2 week1_ass3 week1_ass4 week1_ass5
src.JPG week1_add1.c week1_add2.c week1_ass1.c week1_ass2.c week1_ass3.c week1_ass4.c week1_ass5.c
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ identify src.JPG
src.JPG JPEG 1024x683 1024x683+0+0 8-bit sRGB 322KB 0.010u 0:00.019
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ identify dst.JPG
dst.JPG JPEG 683x1024 683x1024+0+0 8-bit sRGB 112KB 0.000u 0:00.000
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$
```

src.JPG を入力引数で、dst.JPG を出力引数で実行した結果です。イメージの縦と横を入れ換えるためにはイメージを 90 度回転させればよいと思ったので反時計回りで 90 度回転させました。

今回の実習は基本的に C 言語の libgd という画像処理ライブラリを使って行われました。そしてこの問題を含めてこの後の問題のプログラムは下の図みたいなフローで構成されています。



この図の中で詳しく説明をするところは画像処理の部分でこういった処理をしたのかを説明します。

まずこの問題の要求事項は横と縦を入れ換えることです。

それをやるために入力画像を90度曲げました。反時計回りにしました。

元の画像の横と縦の幅をそれぞれWとHとします。出力の画像は縦にはH、横にはWの幅を持ちます。入力画像を f として出力画像を g とすると以下のような式です。

$$g(j, W-1-i) = f(i, j)$$

入力画像の(i,j)のところの画素値を出力画像の(j,W-1-i)のところの画素値にすると上のような結果が出ます。

必須課題 2

必須課題 2 のプログラムの実行方式は必須課題と同じです。

`./week1_ass2 [Input Image] [Output Image]`

この問題の要求事項は入力画像をグレースケールに変えて出力することです。

まず実行結果です。左が入力、右が出力です。

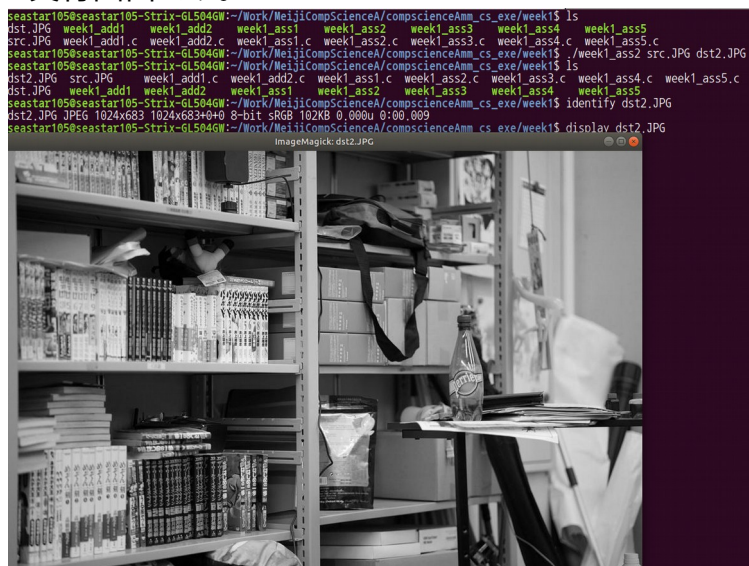


基本的に入力に入る画像は RGB の三チャンネルの画像です。それをグレースケールに変えるためには三つのチャンネルから輝度を計算するべきです。ちょっと難しい話をするると RGB 色空間から YUV 色空間に変えて輝度である Y 値を知る必要があります。そして Y 値を計算する数式は以下のようです。

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

R,G,B は各チャンネルの画素値です。上の数式を使って求めた Y 値を各チャンネルに入れたらグレースケールの画像が出ます。

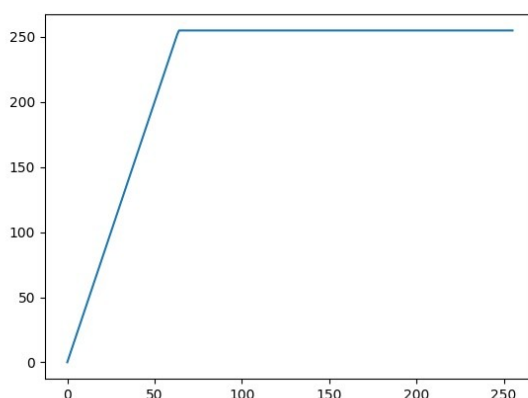
次はターミナルでの実行画面です。



必須課題 3

この課題は傾き 45 度の直線以外の任意のトーンカーブを作って入力画像に適用して出力するプログラムを作成する課題です。

まずトーンカーブとは画素値を入力として画素値を出す関数のことです。この関数の形によって画像に何らかの補正が働きます。下の図は講義資料にあったトーンカーブで全体的に暗い画像を明るく補正してくれます。プログラムでもこのトーンカーブを使いました。



画素値が 64 より小さい時には画素値を四倍して 64 以上の画素値に対しては 255 を出します。このトーンカーブによって画像は明るくなります。

以下は処理前と処理後の画像です。

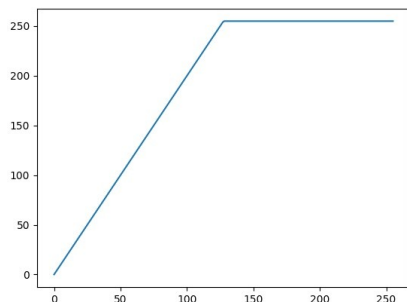


元の画像で暗くてよく見えなかった部分は見やすくなったんですが明るかった部分は見えなくなりました。こういう現象を白飛びとも言います。トーンカーブを少し変えればこの問題はすこしは解消されると思います。

実行コマンドは以下のようです。

`./week1_ass3 [Input Image] [Output Image]`

下の図は変えたトーンカーブとそれを適用した結果です。



以前と変わったところは 64 を基準にするのではなく 128 を基準にただけです。

必須課題 4

この課題は実行時に入力で領域を指定してもらいその領域に対してのみトーンカーブを適用して出力に出すプログラムです。指定した領域が入力ファイルの範囲外だった場合エラーメッセージを出して終了するようになっています。実行コマンドは以下の形式です。

`./week1_ass4 [Input Image] [Output Image] [Starting X Value] [Starting Y value] [Y offset] [X offset]`

次は実行画面とエラーを出したときの画面です。

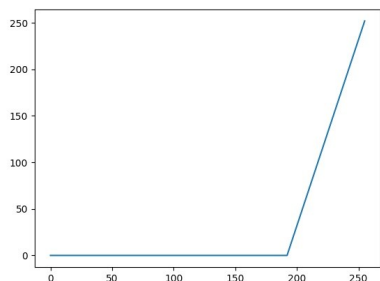
```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ls
dark.JPG  dst3_1.JPG  dst.JPG  week1_add1  week1_add2  week1_ass1  week1_ass2  week1_ass3  week1_ass4  week1_ass5
dst2.JPG  dst3.JPG    src.JPG  week1_add1.c  week1_add2.c  week1_ass1.c  week1_ass2.c  week1_ass3.c  week1_ass4.c  week1_ass5.c
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ./week1_ass4 src.JPG dst4.JPG 100 100 700 300
range error
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ./week1_ass4 src.JPG dst4.JPG 100 100 200 300
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ display dst4.JPG
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$
```

処理前と処理後の画像です。使ったトーンカーブは必須課題 3 と同じです。

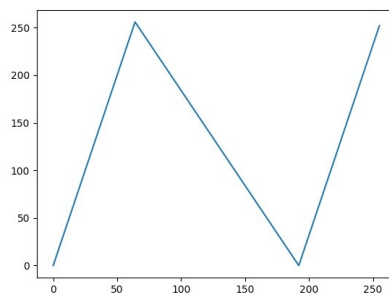


任意課題 1

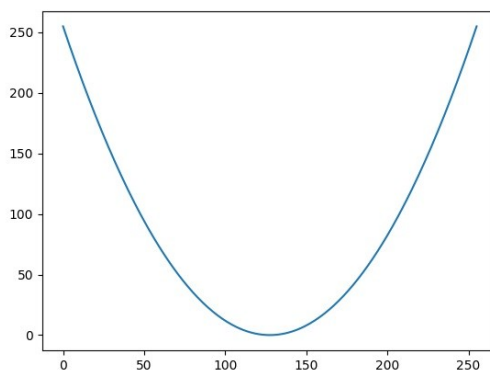
この課題は必須課題 4 と同様トーンカーブの適用領域を入力で指定してもらいその上適用するトーンカーブのナンバーも引数として受けて指定領域とそれ以外の領域にそれぞれ適用するトーンカーブを決めてもらいます。まず指定できるトーンカーブの種類は四つあり必須課題 3 で使ったものが一つあってそれ以外は以下の三つです。



ToneCurve2



ToneCurve3



ToneCurve4

ToneCurve2 は画素値が 192 より小さい場合は 0 を、192 以上(画素値-192)*4 を出します。

ToneCurve3 は画素値を x として

$0 \leq x < 64$	$x * 4$
$64 \leq x < 192$	$(x-192) * (-2)$
$192 \leq x < 256$	$(x-192) * 4$

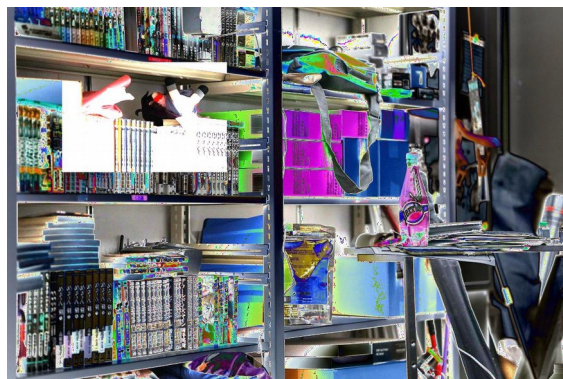
ToneCurve4 は画素値を x として $\frac{(x-127.5)^2}{127.5^2} * 255$ です。本当に任意の関数を設定しました。

指定できるものは四つまでで 0～3 以外の数を入力したらエラーメッセージを出します。以下は実行コマンド形式と実行画面です。

`./week1_ass4 [Input Image] [Output Image] [Starting X Value] [Starting Y value] [Y offset] [X offset]
[Selected area ToneCurve #] [Other area ToneCurve #]`

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ls
dark.JPG dst3_1.JPG dst4.JPG src.JPG week1_add1.c week1_add2.c week1_ass1.c week1_ass2.c week1_ass3.c week1_ass4.c week1_ass5.c
dst2.JPG dst3.JPG dst.JPG week1_add1 week1_add2 week1_ass1 week1_ass2 week1_ass3 week1_ass4 week1_ass5
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ./week1_add1 src.JPG dst4_1.JPG 100 100 200 300 0 4
curve range error
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ./week1_add1 src.JPG dst4_1.JPG 100 100 200 300 0 2
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ display dst4_1.JPG
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$
```


次は処理前と処理後の画像です。



必須課題 5

この課題は画像検索の一番基本であるテンプレートマッチングを具現するものです。検索対象の画像の中に探そうとするテンプレート画像が含まれているかを判定しあると判定したら検索対象の画像にテンプレート画像がある位置に四角形として表示した画像を出力として出します。

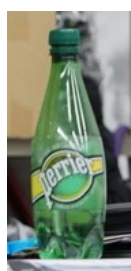
検索に使う方法はテンプレート画像が検索対象の画像に位置できるすべての位置に対してテンプレート画像の全部の座標の画素値に対して検索対象でその座標に対応する座標の画素値との差を二乗して全部足し算します。その誤差の足したものが閾値以下だったらテンプレート画像が検索対象の中に入っていると判定します。

次はプログラムの実行コマンドです。

```
./week1_ass5 [Searching Image] [Template Image] [Output Image] (Threshold)
```

閾値は第4引数で指定できますが、指定値がない場合プログラム内に先に指定した閾値を使います。閾値より低い位置が多数あった場合誤差の足したものが一番低い位置に四角形を表示して出力します。

以下は処理前の画像と処理後の画像です。



Template



Searching Image



Output Image

この方式の問題点ですが、まずすべての座標に対して計算を行うので時間がかかりかかります。そのためこのプログラムでは誤差の合が閾値を越えた瞬間次の位置に対して新に計算を始めることで少しは時間を減らしました。

そして方法自体の問題ですが、テンプレート画像の画素値を直接に検索対象の対応座標の画素値と比べるためテンプレート画像の大きさが少し変わることで見つけられなくなります。これはパターン認識にはほとんど使えないのはこの理由です。

任意問題 2

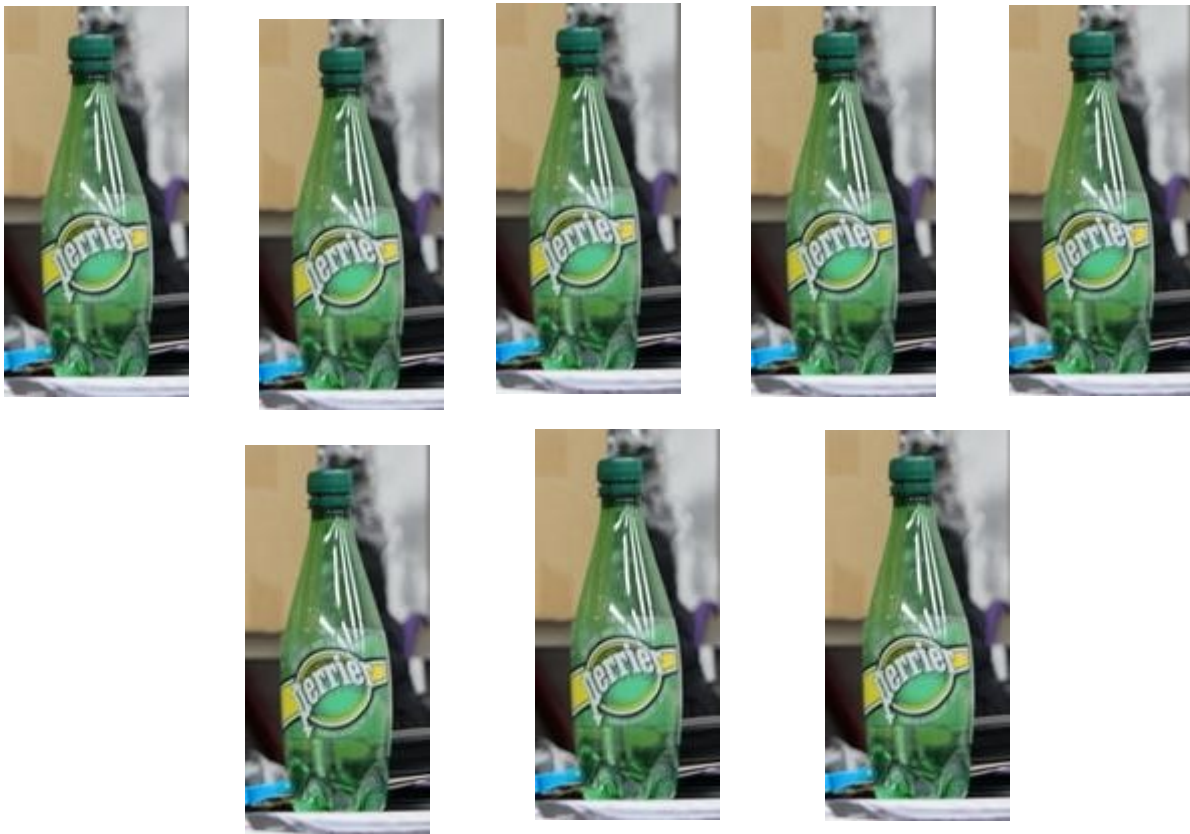
必須課題 5 では閾値以下の座標の中で一番小さい位置しか出力に足さないんですが、この問題では閾値より低いすべての位置を出力として出すよう変えました。実行時に第 3 引数で出力ファイル名のプレフィックスを受けます。実行コマンドは以下の形式です。

`./week1_add2 [Searching Image] [Template Image] [Prefix] (Threshold)`

閾値より低い部分だけを出力に出すため libgd の `gdImageCrop` 関数を使いました。

以下は必須課題 5 と同じ入力を使った結果に出たものです。

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ls
dark.JPG  dst3_1.JPG  dst4_1.JPG  dst5.JPG  src.JPG  week1_add1  week1_add2  week1_ass1  week1_ass2  week1_ass3  week1_ass4  week1_ass5
dst2.JPG  dst3.JPG   dst4.JPG   dst.JPG   template.JPG  week1_add1.c  week1_add2.c  week1_ass1.c  week1_ass2.c  week1_ass3.c  week1_ass4.c  week1_ass5.c
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ./week1_add2 src.JPG template.JPG output
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$ ls
dark.JPG  dst3.JPG  dst5.JPG  output0001.JPG  output0004.JPG  output0007.JPG  template.JPG  week1_add2  week1_ass1.c  week1_ass3  week1_ass4.c
dst2.JPG  dst4_1.JPG  dst.JPG   output0002.JPG  output0005.JPG  output0008.JPG  week1_add1  week1_add2.c  week1_ass2.c  week1_ass3.c  week1_ass5.c
dst3_1.JPG  dst4.JPG  output0000.JPG  output0003.JPG  output0006.JPG  src.JPG  week1_add1.c  week1_ass1  week1_ass2.c  week1_ass4  week1_ass5.c
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week1$
```



上記のイメージが output0000 から output0008 までの結果です。

参考文献

コンピュータサイエンス実習 A(マルチメディア) 第 1 回画像処理の基礎 (講義資料)