

形&大きさ変換

米田 幸生



(a) 元画像



(b) 反転後

ネガ・フィルムのように見える

図1 反転処理1…色の反転

画像の形や大きさを変える処理を紹介します。M5 Stackの画面は320×240画素と大きくはないので、画面を有効利用するために、必要なところを拡大して切り取って表示することが求められます。

本誌で連載していた「IT農家のディープ・ラーニング奮闘記」では、きゅうり画像の学習用データを増やすために、画像の拡大、縮小、回転、上下反転など、まさにここで紹介するような技術を利用していました。

リスト1 色の反転

```
#ifndef REVERSECOLOR_H
#define REVERSECOLOR_H

#include "Original.h"

class ReverseColor : public Original {

public:
    virtual void display(const char* fname);

};

#endif // REVERSECOLOR_H
```

Original 画像処理クラスを継承して色反転クラスを作っている。以降のほとんどの画像処理クラスで、同様の構成になっている

(a) ヘッダ(ReverseColor.h)

```
#include "ReverseColor.h"

void
ReverseColor::display(const char *fname){
    uint8_t *pImg;
    uint8_t *pImg_disp;
    int x,y,bx,by;

    JpegDec.decode(fname);

    while(JpegDec.read()){
        pImg = JpegDec.pImage;
        pImg_disp = pImg;

        for(by=0; by<JpegDec.MCUHeight; by++){
            for(bx=0; bx<JpegDec.MCUWidth; bx++){
                *pImg = 255 - *pImg;
                *(pImg + 1) = 255 - *(pImg + 1);
                *(pImg + 2) = 255 - *(pImg + 2);
                pImg += 3;
            }
        }
        x = JpegDec.MCUx * JpegDec.MCUWidth;
        y = JpegDec.MCUy * JpegDec.MCUHeight;
        displayPixel(x, y, JpegDec.MCUWidth,
                    JpegDec.MCUHeight, pImg_disp);
    }
    JpegDec.finish();
}
```

画素値の最大値(255)から引くことで反転している

(b) メイン(ReverseColor.cpp)

画像の反転

● 色の反転

取得したRGB値をRGBそれぞれについて最大値(255)から引いた値にすることで、色を反転させるこ