

3年15組92番 全 海成(JEON HAESUNG)

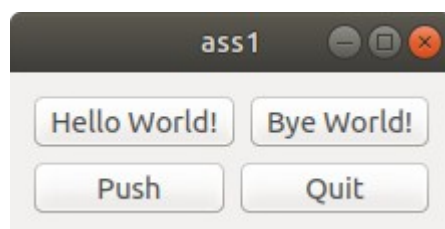
コンピュータサイエンス実習 A マルチメディア 四回目 レポート 4

今回の実習は Qt4 を使って GUI プログラムを作りました。それと OPENCV を使って画像処理を行った後その結果を GUI プログラムとして表すのが目的です。

### 必須課題 1

この課題は Qt を使って四つのボタンを 2 X 2 の形で持つプログラムを書くことです。この際各ボタンは何らかのイベントに対応します。

書いたプログラムを実行した結果は下のようなウィンドウです。



Hello World というボタンを押せば hoge がコンソールに表示されます。

Bye World ボタンが押されたら hoge2 がコンソールに出力されます。

Push ボタンが押されたら Now Japanese Unavailable がコンソール上に出力されます。

Quit ボタンが押されたらプログラムが終了します。

ボタンを押したとき実行結果です。

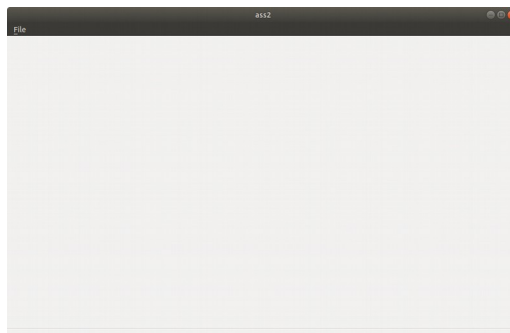
```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week4/ass1$ !.  
./ass1  
Gtk-Message: 16:54:43.764: Failed to load module "canberra-gtk-module"  
hoge  
hoge  
hoge2  
hoge2  
Now Japanese Unavailable  
Now Japanese Unavailable  
Quit  
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week4/ass1$
```

プログラムは ./ass1 で実行できます。

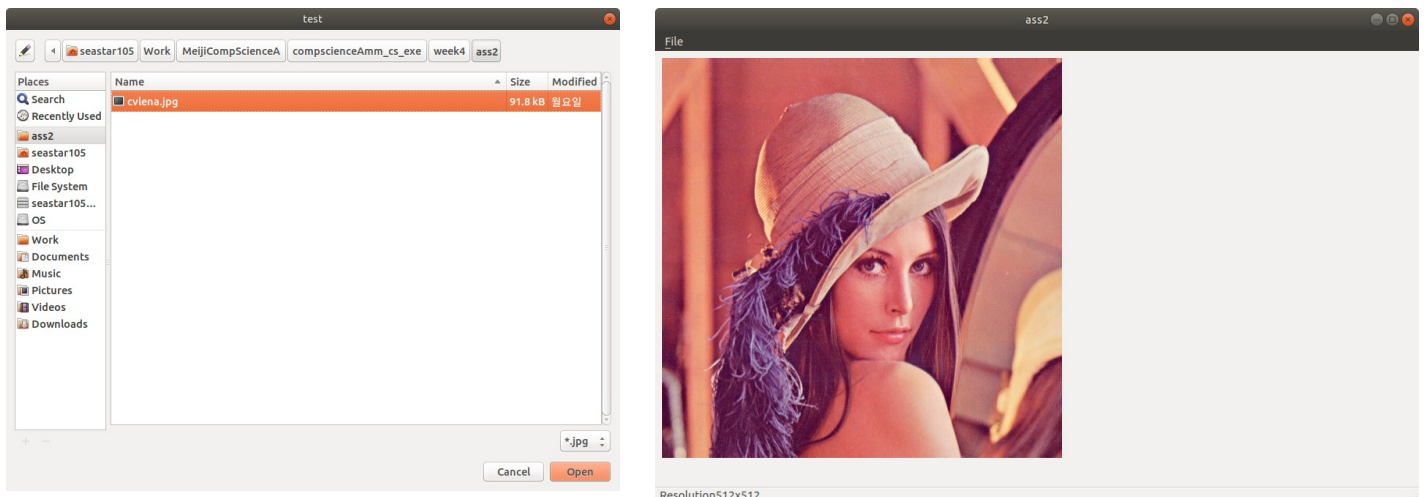
### 必須課題 2

この課題は Qt と OPENCV を使います。Qt で提供する画像を扱うデータ構造として

QImage がありますがこの QImage と OPENCV で画像を扱う Mat というデータ構造が画素値の保存方法が違います。そのため OPENCV で画像ファイルを読み込んで Qt に出力するためには適切な変換が必要となります。この課題で書いたプログラムをまさにそのことをやるプログラムです。Qt のダイアログボックスを使って画像ファイルを指定し OPENCV を使ってその画像ファイルを読み込み Qt を使って画面に出力します。下はプログラムの実行コマンドと実行して最初に出る画面です。



File をクリックして Load をクリックするとダイアログボックスが出て何のファイルを読み込むのかを指定します。そしたらその画像ファイルを表示します。



下は opencv を使って画像を開き QImage に変えるコードです。

```
void MyWindow::load()
{
    if(!curFile.isEmpty()) {
        if(QFile::exists(curFile)) {
            cv::Mat m = cv::imread(curFile.toStdString().c_str(),1);
            QImage im(m.data,m.cols,m.rows,QImage::Format_RGB888);
            im = im.rgbSwapped();
            int sWidth, sHeight;
            sWidth = mWidth = im.width(); sHeight = mHeight = im.height();
            if(sWidth > qApp->desktop()->width()/2) sWidth = qApp->desktop()->width()/2;
            if(sHeight > qApp->desktop()->height()/2) sHeight = qApp->desktop()->height()/2;
            mImage = im.scaled(sWidth,sHeight,Qt::KeepAspectRatio);
            setMinimumSize(mImage.size());
            update();
        }
    }
}
```

File の Load が押されたら MyMainWin の load が呼ばれ load の中でダイアログボックスを開いてファイルの名前を得た後に MyWindow の SetFileName が呼ばれてそこで上にある MyWindow の load が呼ばれます。

### 必須課題 3

この課題は今までに学んだ機能を自由に組み合わせ何らかの画像処理を行うプログラムを作成する事です。僕が作成したのは引数として動画ファイルや USB カメラをもらってそれを Qt に出力し四つのボタンを用意して流れている動画に対して行う画像処理を選び適用するプログラムです。

適用できる画像処理は左右反転とガウシアンブラーとエッジ検出です。すべての画像処理は opencv で提供する関数を使いました。まず実行コマンド形式です。

./ass3 [Video Source]

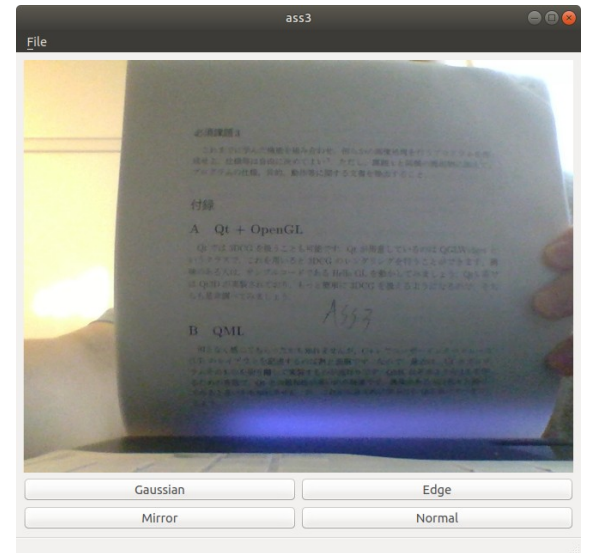
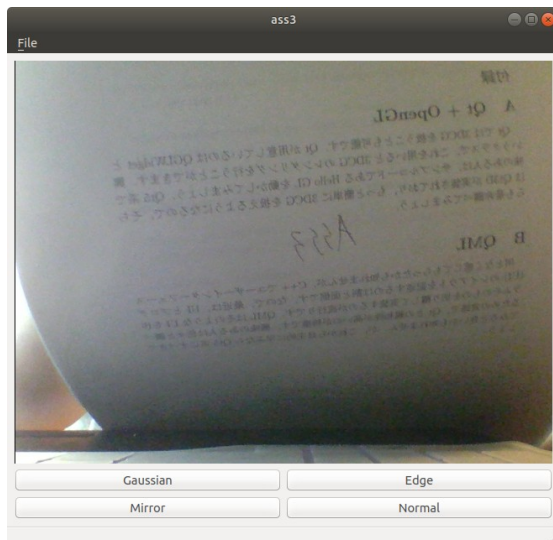
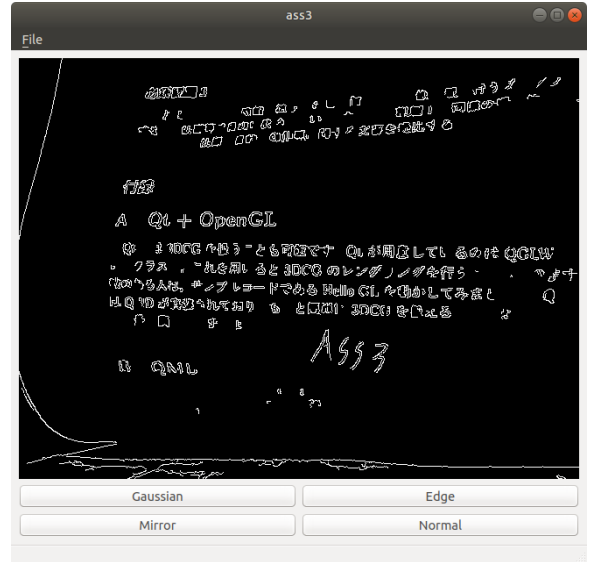
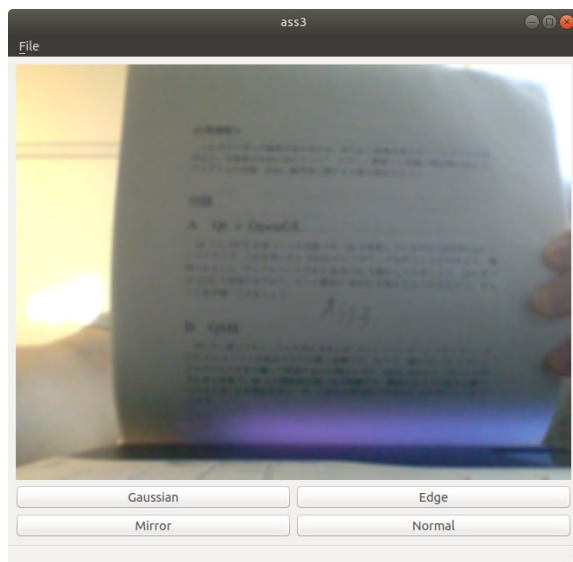
Gaussian を押したらガウシアンブラーが適用されます。

Edge を押したらエッジ検出を行った画面が出力されます。

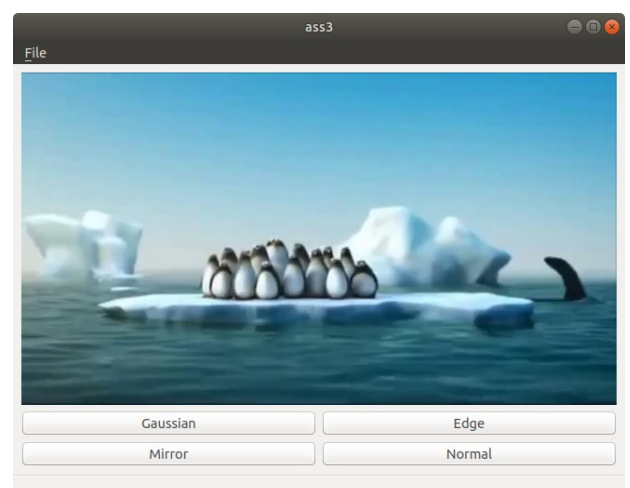
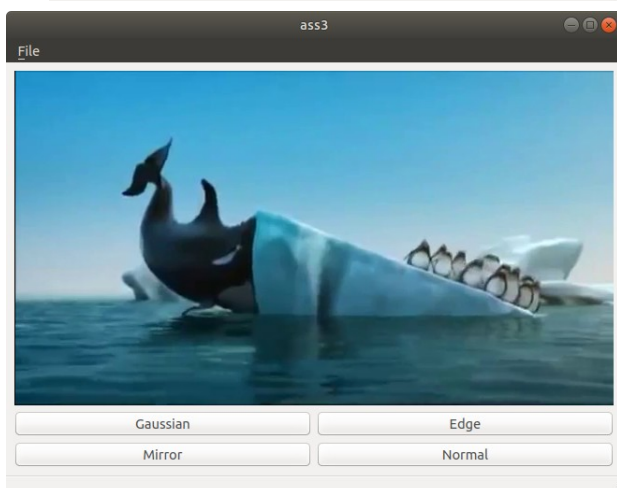
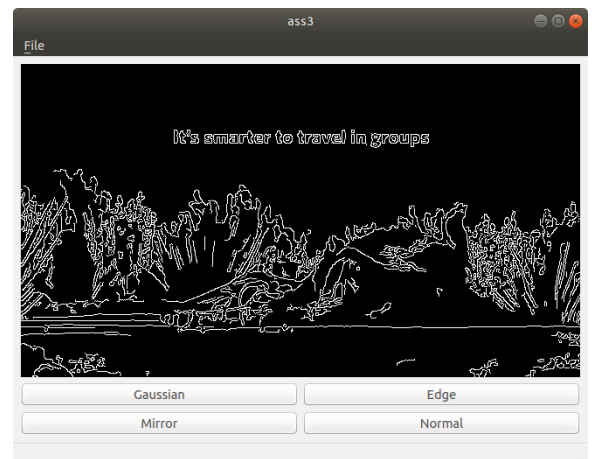
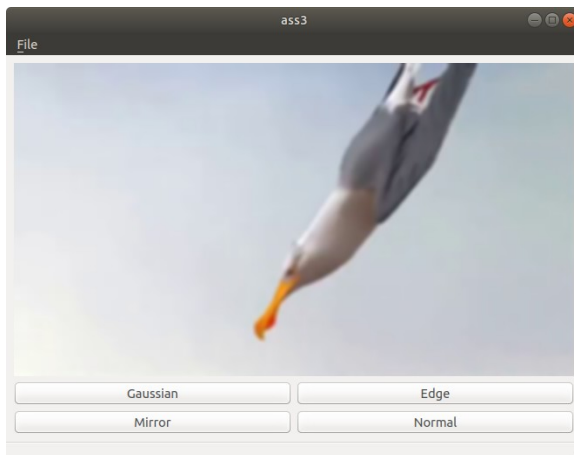
Mirror は左右反転された画像が出力されます。

Normal は元画像を出力します。

下は USB カメラを引数で渡した時の実行画面です。



下は動画ファイルを引数で渡したときです。





各ボタンは MyMainWin で管理されており画像を出力しているところは MyWindow です。各ボタンが押されたら MyWindow にシグナルを送って各シグナルに合わせて現在の処理タイプを変えます。下は MyWindow の DrawVideo 関数でこの間数の中で OPENCV で画像処理を行い出力する QImage に変えます。

```
void MyWindow::drawVideo() {
    Mat frame,gray;
    mCap >> frame;
    switch(curType) {
        case normal: out = frame; break;
        case gauss:
            GaussianBlur(frame,out,Size(7,7),2.5);break;
        case edge:
            Canny(frame,out,50,150);
            break;
        case mirror:
            warpAffine(frame,out,mirrorMat,out.size());break;
        default:
            break;
    }
    if(curType == edge) {
        QImage img((uchar*)out.data, out.cols, out.rows, QImage::Format_Indexed8);
        img.setColorTable(sColorTable);
        mImage = img;
    }
    else {
        QImage img(out.data, out.cols, out.rows, QImage::Format_RGB888);
        mImage = img.rgbSwapped();
    }
    update();
}
```

下は現在の処理タイプを変えるために作った簡単な関数です。タイプは enum で定義されています。

```
void MyWindow::Gaussian() {
    curType = gauss;
}

void MyWindow::Edge() {
    curType = edge;
}

void MyWindow::Mirror() {
    curType = mirror;
}

void MyWindow::Normal() {
    curType = normal;
}
```