

# 數位系統導論實驗

## Lab3 DNN-based Handwritten Digit Recognition

負責助教：王偉丞

Email: [wmike851223@gmail.com](mailto:wmike851223@gmail.com)

## 課程目標

1. 基於Lab1(以MNIST Dataset與Tensorflow訓練DNN)與Lab2(以C完成DNN編程)，完成手寫數字辨識計算核心
2. 利用Qt開發簡易的UI，完成手寫數字辨識系統(如下圖)



## Qt Framework

- Qt是跨平台的C++軟體開發套件，具有非常好的可移植性、跨平台支援，也擁有完善的圖形函式庫，經常被用來開發圖形介面程式。
- Qt能夠用來設計使用者介面，其應用相當地廣泛，像是被Disney用來製作部分的3D動畫、做為TESLA汽車中的套件、醫療用的UI及心電圖等。

## 環境設置 - 在Linux作業系統使用Qt

■ 本課程範例在Linux進行，在此介紹如何架設虛擬環境以在Windows執行Linux

1. 安裝Oracle VirtualBox
2. 下載Ubuntu 16.04
3. 點選新增並設定OS類型/記憶體大小/虛擬硬碟
4. 設定值 > 存放裝置 > 選擇虛擬光碟
5. 啟動
6. 安裝Ubuntu 並重新啟動

(教學影片：<https://youtu.be/F4AnbIWRwwU>)

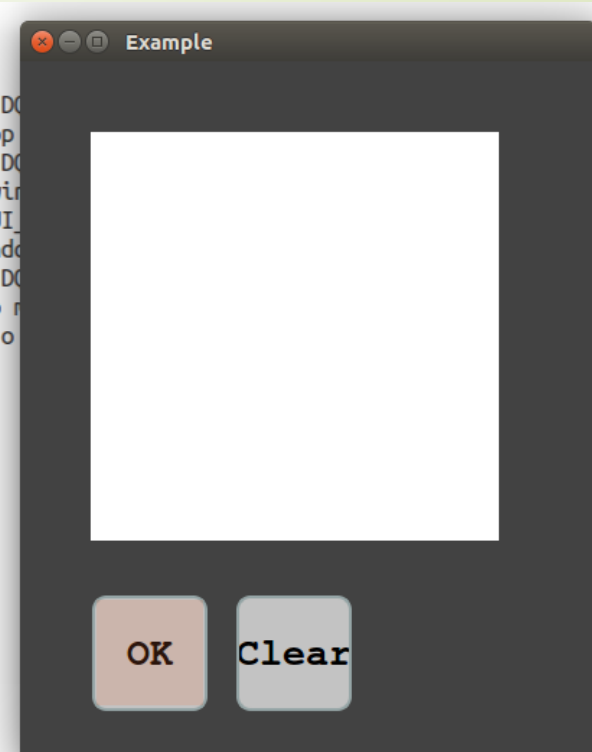
■ 安裝Linux後，開啟終端機輸入以下指令安裝Qt(安裝後參考下頁執行範例，有顯示繪圖介面及代表成功)

- `sudo apt install qt4-qmake`
- `sudo apt install libqt4-dev`

## 範例執行步驟

1. 在example資料夾按右鍵並開啟終端機
2. 輸入指令“qmake-qt4 -project”生成專案
3. 輸入指令“qmake-qt4”生成Makefile
4. 輸入指令“make”以執行Makefile來編譯程式
5. 輸入指令“./example”執行程式

```
deus@deus:~/Downloads/lab3/example$ qmake-qt4 -project
deus@deus:~/Downloads/lab3/example$ qmake-qt4
deus@deus:~/Downloads/lab3/example$ make
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI -I/usr/include/qt4 -I. -I. -o main.o main.cpp
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI -I/usr/include/qt4 -I. -I. -o mywindow.o mywindow.cpp
/usr/lib/x86_64-linux-gnu/qt4/bin/moc -DQT_NO_DEBUG -DQT_GUI -I/usr/include/qt4 -I. -I. mywindow.h -o moc_mywindow.o
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI -I/usr/include/qt4 -I. -I. -o moc_mywindow.o moc_mywindow.o
g++ -m64 -Wl,-O1 -o example main.o mywindow.o moc_mywindow.o
deus@deus:~/Downloads/lab3/example$ ./example
[]
```



## 課程範例 - 繪圖介面

- 在本範例中，同學可藉由如何用Qt製作視窗介面、繪圖框與按鈕，學習Qt的基本應用。
- 包含繪圖、圖片儲存與清除畫框功能，本課程將程式分成視窗設計、按鈕功能與繪圖框三個部分，逐一與同學介紹。





## 繪圖介面 - 視窗設計 (1/5)

### ► 利用QWidget設計簡單視窗

```
3  /* Class Constructor */
4  MyWindow::MyWindow(QWidget *parent) : QWidget(parent)
5  {
6      // 初始化變數
7      stroke.clear();
8      lines.clear();
9      painting = false;
10
11     // 主視窗
12     setWindowTitle("Example");
13     resize(400, 480);
14     setStyleSheet("background-color:#424242;");
```

### ► 可參考[官方文件](#)，加入自己喜歡的功能

## 繪圖介面 - 按鈕及其樣式 (2/5)

### ► 利用QPushButton建立按鈕並設計其樣式

```
16  /* OK 按鈕 */
17  conf_btn = new QPushButton("OK",this); //按鈕名字
18  conf_btn->setStyleSheet(
19      "QPushButton{border: 2px solid #95A5A6;border-radius:10px;color:rgb(0,0,0);background-color:rgb(195,195,195)}"); //按鈕樣式
20  conf_btn->setFont(QFont("Courier", 20, QFont::Bold)); //按鈕字體
21  conf_btn->setGeometry(50, 370, 80, 80); //按鈕位置及大小
22  connect(conf_btn, SIGNAL(clicked()), this, SLOT(ConfirmButton_clicked())); //點擊按鈕後會呼叫ConfirmButton_clicked函式
23
24  /* Clear 按鈕 */
25  clr_btn = new QPushButton("Clear",this);
26  clr_btn->setStyleSheet(
27      "QPushButton{border: 2px solid #95A5A6;border-radius:10px;color:rgb(0,0,0);background-color:rgb(195,195,195)}");
28  clr_btn->setFont(QFont("Courier", 20, QFont::Bold));
29  clr_btn->setGeometry(150, 370, 80, 80);
30  connect(clr_btn, SIGNAL(clicked()), this, SLOT(ClearButton_clicked()));
31 }
```



## 繪圖介面 - 按鈕功能 (3/5)

### ■ 點擊按鈕所呼叫的功能函式

```
33  /* 將繪圖框存成jpg */
34  void MyWindow::ConfirmButton_clicked()
35  {
36      // 選取存圖範圍
37      QPixmap qpx;
38      QImage img = qpx.grabWidget(this, 50, 50, 280, 280).toImage();
39
40      img.save(img_name, "jpg");
41      update();
42  }
43
44  /* 清除畫框 */
45  void MyWindow::ClearButton_clicked()
46  {
47      stroke.clear(); // 清除當前軌跡
48      lines.clear();  // 清除畫過的位置
49      painting = false;
50      update();
51  }
```

# 繪圖介面 - 繪圖功能 (4/5)

## ➡ 將當前繪圖軌跡顯示與儲存在畫框上

```
53 void MyWindow::mousePressEvent(QMouseEvent* event)
54 {
55     /* 畫筆只能落在此範圍: 325>x>55 && 325>y>55 */
56     if(event->x()>55 && event->x()<325 && event->y()>55 && event->y()<325)
57     {
58         painting = true;
59         stroke.push_back(event->pos()); //存下落筆座標
60     }
61     update();
62 }
63
64 void MyWindow::mouseMoveEvent(QMouseEvent* event)
65 {
66     /* 只有在滑鼠按下後才紀錄鼠標移動過的座標 */
67     if(event->x()>55 && event->x()<325 && event->y()>55 && event->y()<325 && painting)
68     {
69         stroke.push_back(event->pos());
70         update();
71     }
72 }
```

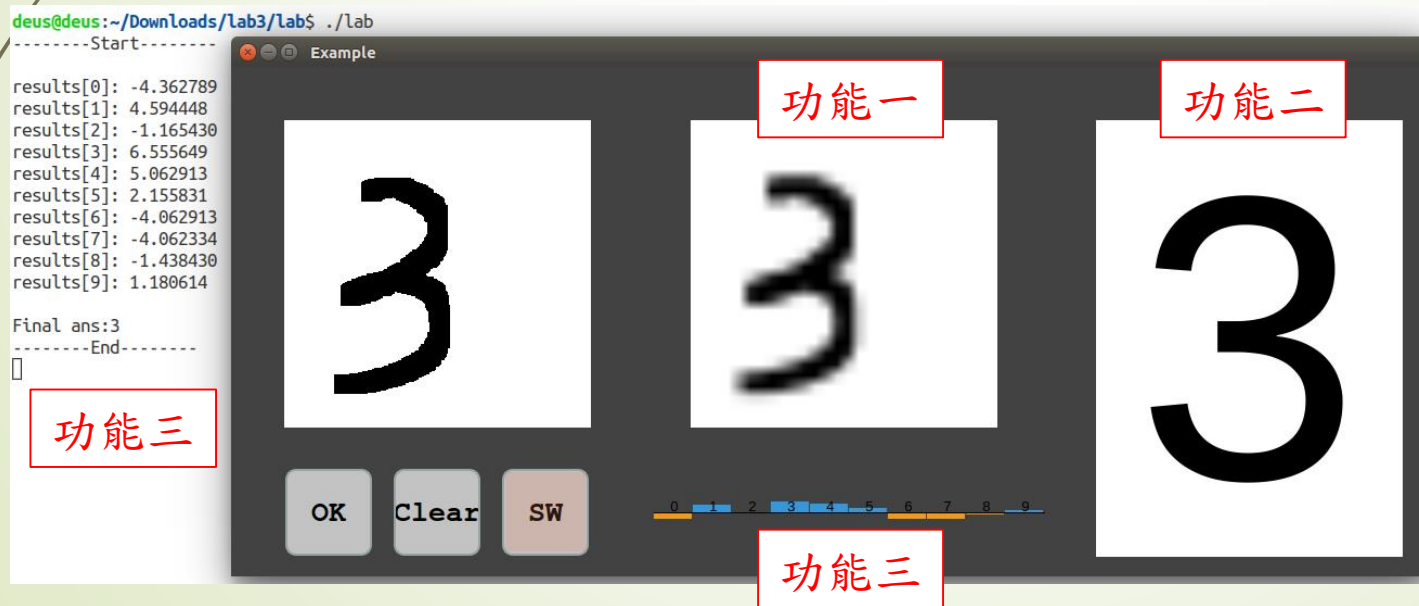
```
74 void MyWindow::mouseReleaseEvent(QMouseEvent*)
75 {
76     /* 按下滑鼠後鬆開會將畫過的軌跡存在畫布上，並清空軌跡 */
77     if(painting)
78     {
79         painting = false;
80         lines.push_back(stroke);
81         stroke.clear();
82     }
83     update();
84 }
```

## ► 提供繪圖功能的畫框設置

```
86 void MyWindow::paintEvent(QPaintEvent*)
87 {
88     // 設定畫筆樣式
89     QPainter painter(this);
90     painter.setFont(QFont("Arial", 10));
91     painter.setBrush(Qt::white);
92     painter.setPen(QPen(QColor("#FFFFFF")));
93
94     // 建立畫框範圍
95     painter.drawRect(49, 49, 282, 282);
96
97     // 設定畫筆樣式
98     painter.setPen(QPen(Qt::black, 18));
99
100    // 繪出畫過的位置
101    for(int i=0; i<lines.size(); i++) {
102        for(int j=0; j<lines[i].size(); ++j)
103            painter.drawPoint(lines[i][j].x(), lines[i][j].y());
104    }
105
106    // 繪出當前落筆後的軌跡
107    for(int i=0; i<stroke.size(); i++)
108        painter.drawPoint(stroke[i].x(), stroke[i].y());
109 }
```

# LAB - 手寫數字辨識系統

- 題目：完成手寫數字辨識的使用者介面與結合DNN模型，有下列三個功能
- 功能一：將按下OK按鈕後儲存的圖改成用解析度為28x28的ppm檔儲存，並顯示在中間圖框
  - 功能二：按下SW按鈕後，執行Lab2中以C完成的DNN模型，將功能一儲存的圖做為DNN輸入，進行數字辨識並將辨識結果繪在右側圖框
  - 功能三：輸出DNN中Output Layer的10個Neurons值並以直方圖表示



## 課程評分

- Demo時間 & 梯次：參照分組名單上個人Demo時間
- Demo地點：工一館206
- 評分方式：
  - (1) 範例成功執行 30%
  - (2) Lab的三個功能各佔15%
  - (3) 隨堂練習 25%

## 附錄－學習資源

- 阿洲的程式教學

<http://monkeycoding.com/>

- OPENHOME

<https://openhome.cc/Gossip/Qt4Gossip/index.html>

- Qt 官網

<http://doc.qt.io/qt-4.8/>

- Qt 指南

<https://qtguide.ustclug.org/>

- Qt 學習之路

<https://www.devbean.net/2012/08/qt-study-road-2-catalog/>