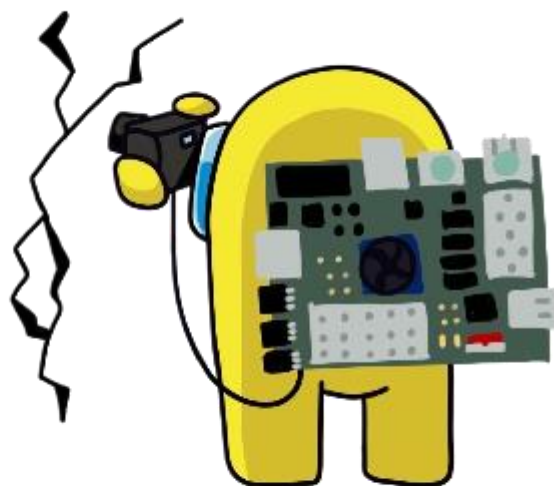


# 人工智慧 FPGA 邊緣運算系統 (以瑕疵辨識為例)

## 成果報告書



指導教授：陸子強、林浩仁

實驗室名稱：智慧感知與互動技術實驗室

專題學生：許嘉顯、李以歲、何奐文

陳鍊翔、王昱閔、蒙羿辰

# 目錄

1. 前言	3
2. 研究動機	3
3. 開發工具	4
I. Pytorch	
II. Vitis AI	
III. Petalinux	
IV. Vivado	
V. PyQt	
VI. QT Designer	
VII. Zynq UltraScale + MPSoC ZCU104	
4. 規格內容	6
I. 模型	
II. 電路圖	
III. UI 介面	
5. 關鍵技術	8
I. 模型量化	
II. 模型簡化	
6. 系統架構	9
7. 經費預算	10
8. 分工表格	11
9. 結論及未來展望	11

## 前言

近年來人工智慧越來越於各大場合出現，當中不乏有許多使用人工智慧的場合有硬體上的限制，包括供電、效能等等，又或者是如全世界所推行的環保也會有影響。而 FPGA 是一種可以自己設計電路輸出入的邊緣運算裝置，正因為如此，FPGA 相較於其他邊緣運算裝置可以有更出色的效能能耗比，因此本專題之主軸使用了 FPGA 作為邊緣運算裝置，在此裝置上運行深度模型，並且本專題以瑕疵檢測為例，針對三種不同的焊錫範例進行分類。

## 研究動機

- 我們也希望可以以 FPGA 做到與普通電腦一樣的執行效率，以此來降低成本，並且 FPGA 也相較於個人電腦更貼近工廠當中工業電腦的系統。
- 現今的大多數人工智慧工廠都搭配了雲端運算的方式來推論深度模型，這種做法會有資訊安全的疑慮，而如果使用邊緣運算，則可以讓數據不用藉由雲端也能完成運算。

## 開發工具

### 2.1 Pytorch

Pytorch 是一個開源的深度學習框架，它提供了強大的工具和資源，讓開發者能夠建立並訓練深度神經網路，並且 Pytorch 可以利用 GPU 進行加速，這對於處理大規模數據和複雜的模型特別有益。Pytorch 的 CUDA 支援使得在 GPU 上訓練模型變得相對簡單。

### 2.2 Vitis AI

Vitis AI 是由 Xilinx（賽靈思）提供的一個深度學習推理工具套件，專為 Xilinx 的可重構計算架構（Reconfigurable Compute Fabric）裝置設計。Vitis AI 旨在使開發者能夠在 Xilinx FPGA（可編程閘陣列）上部署和運行深度學習模型，從而實現高性能的深度學習推理。

### 2.3 Petalinux

PetaLinux 是 Xilinx 提供的一個嵌入式 Linux 系統開發工具套件，專為在 Xilinx 系列可編程邏輯裝置（PL）和處理器系統（PS）上運行的嵌入式應用程式提供支援。PetaLinux 的目標是簡化嵌入式 Linux 系統的建構、配置和部署過程。

### 2.4 Vivado

Vivado 是 Xilinx 提供的一個綜合的可編程邏輯裝置（PL）和處理器系統（PS）設計工具套件。它主要用於設計和開發基於 Xilinx FPGA 的數字電路和系統。

### 2.5 PyQt

PyQt 是一個用於 Python 語言的 GUI 開發工具組，它基於 Qt 框架。Qt 是一個跨平台的應用程序和 GUI 開發框架，並提供了用於構建現代且具有吸引力的用戶界面的工具和庫。PyQt 就是將 Qt 的功能引入 Python 開發環境中，使得開發者可以使用 Python 語言來建構強大的跨平台應用程式。

## 2.6 Qt Designer

Qt Designer 是一個可視化的界面設計工具，允許開發者在視覺化環境中構建和設計界面。開發者可以通過拖放操作將元件放置在視窗中，然後通過屬性編輯器進行進一步的設置。

## 2.7 Zynq UltraScale + MPSoC ZCU104

可進行邏輯編程的開發板。



圖 2-1

## 規格內容

### 3.1 模型

原先使用 VGG16 模型，但模型的泛化能力欠佳，因此本專題將模型加上 BN 層，並把兩層全連結層改為全域平均池化。

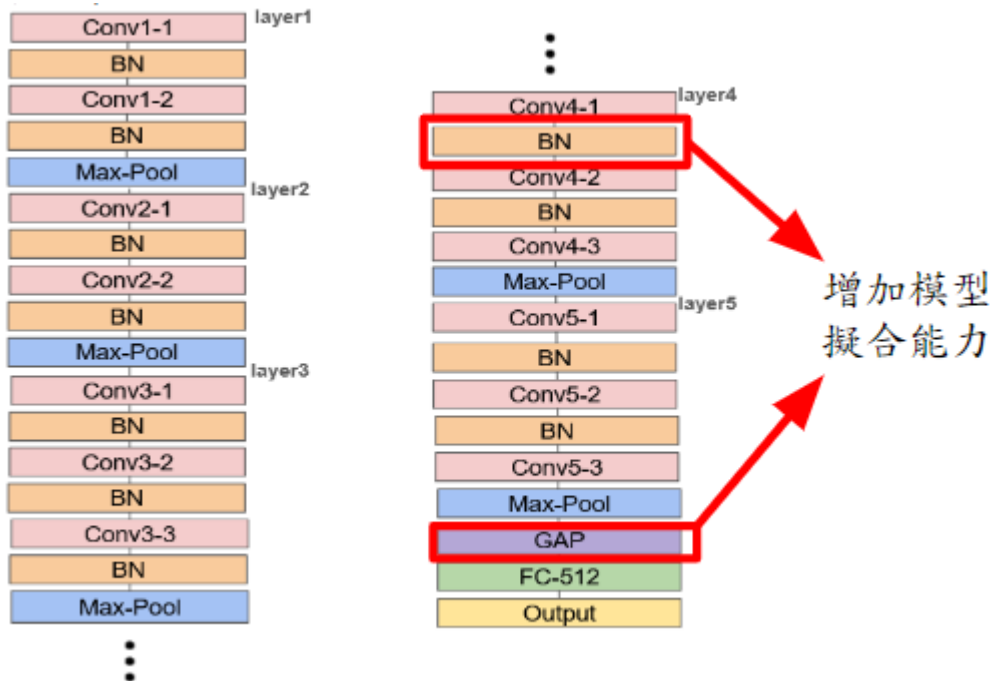


圖 3-1

### 3.2 電路圖

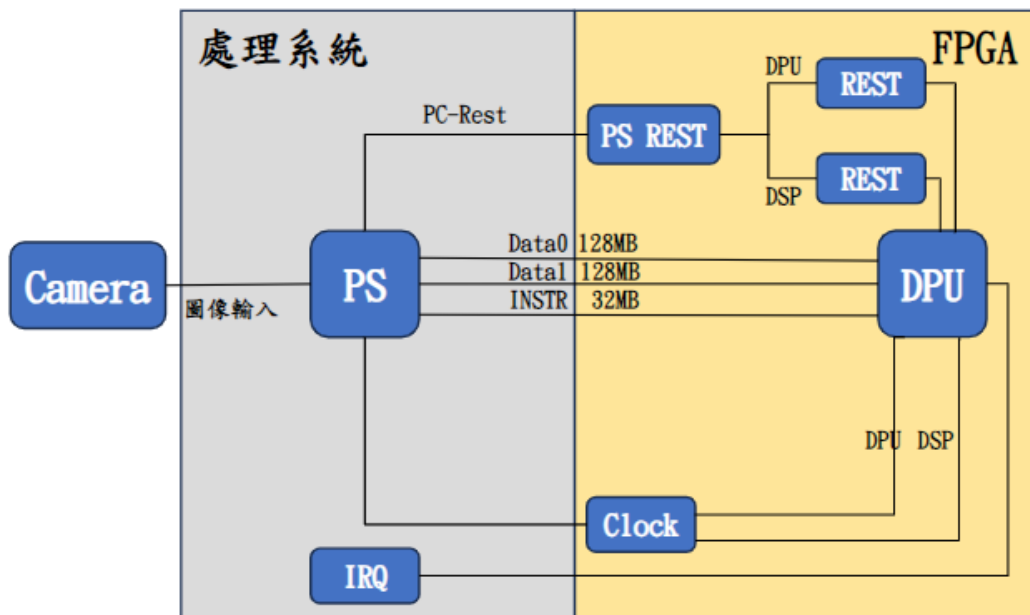


圖 3-2

### 3.3 UI 介面

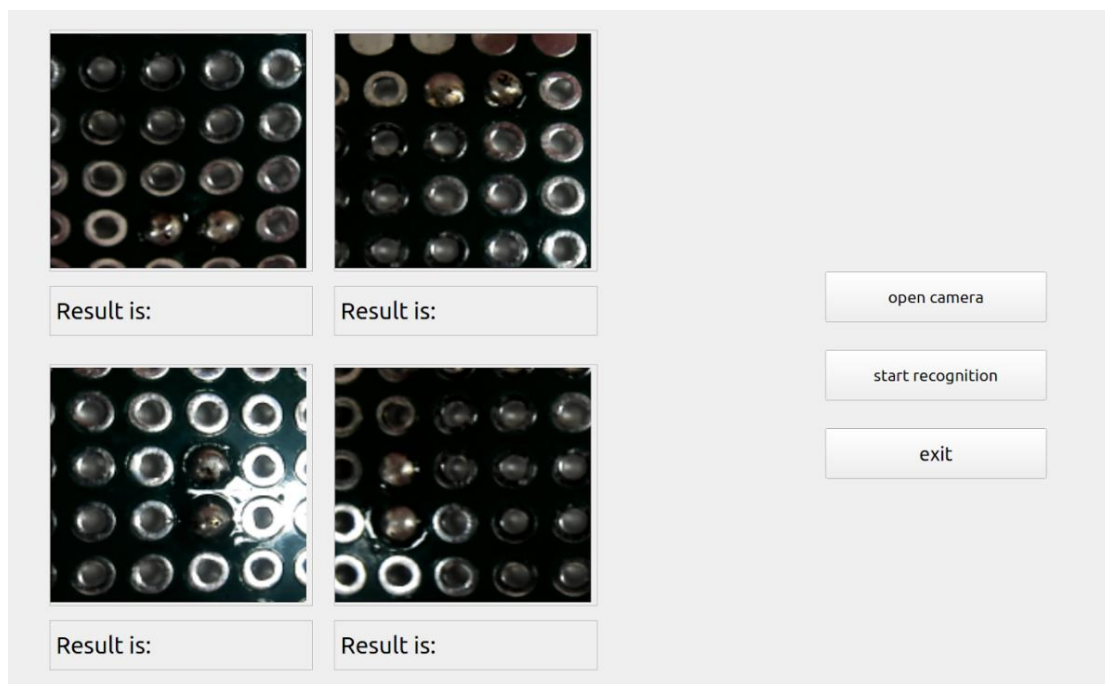


圖 3-3

## 關鍵技術

### 4.1 模型量化

模型量化是指將深度學習模型的權重和特徵映射（activation）轉換為更小的數據表示的過程。這個過程的目的是減小模型的體積，降低模型的計算和記憶體需求，同時在一定程度上保持模型的準確度。這對於在此專題這種資源受限的環境中部署深度學習模型非常重要。

### 4.2 模型簡化

通過簡化模型的結構，例如減少層數、神經元數量或使用更簡單的模型結構，以減小模型的複雜度和大小。



## 系統架構

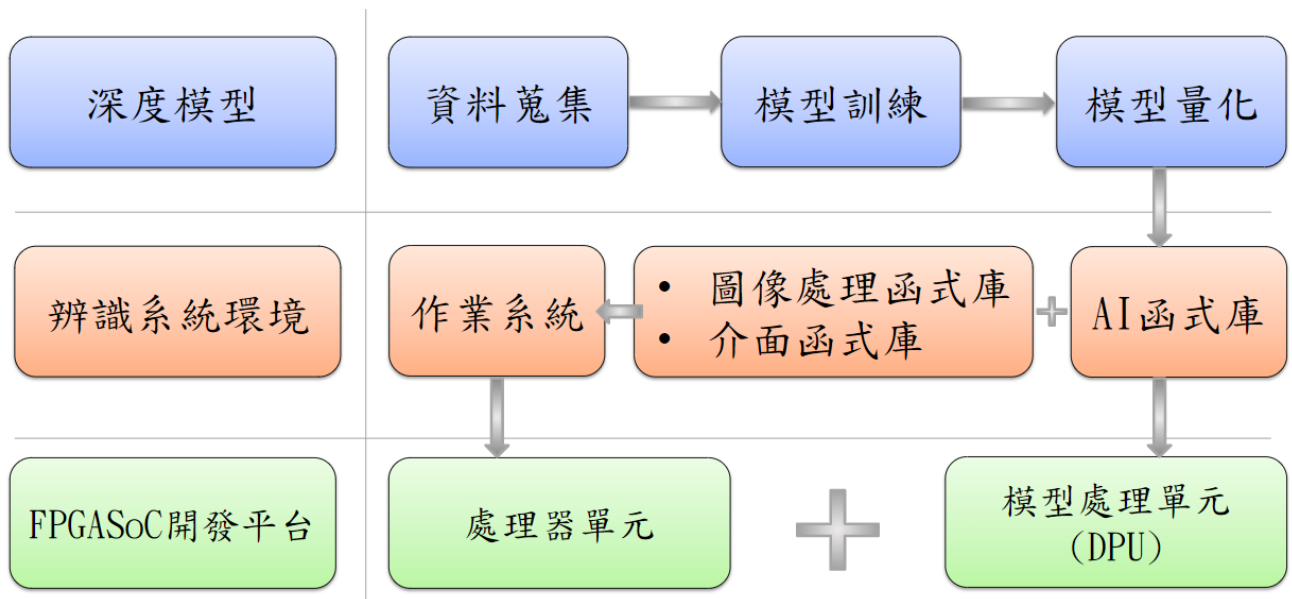


圖 5-1

## 經費預算表

備註：單人開發成本皆為  
 時薪176(元)\*每日8(小時)\*每月22(天)=30976(元)  
 各人力開發項目為月薪\*開發時長

## 成本結算表

項目	金額(元)	開發時長(月)
FPGA開發板	45000	無
模型訓練設備	80000	無
硬體加速開發人力成本	30976	1個月
人工智慧開發人力成本	模型訓練： $30976*3=92928$	3個月
	模型量化 $30976*2=61952$	2個月
作業系統開發人力成本	30976	1個月
使用者介面開發人力成本	$30976*2=61952$	2個月
預估金額	403784(元)	

圖 6-1

## 分工表格

### 分工表

姓名	工作內容
李以歲	模型訓練、資料蒐集
許嘉顯	模型訓練、資料蒐集
何煥文	模型量化
蒙羿辰	硬體設計
王昱閔	嵌入式系統設計
陳鏈翔	GUI介面設計

圖 7-1

### 未來展望

#### 一、 資料的再蒐集和模型再訓練

因為這次模型的訓練資料數量較少，所以未來如再增加資料量可以再增加模型的準確度。

#### 二、 Camera 輸入的 FPS 有上限

我們目前用的 camera 的輸入上限為 30FPS，但開發版能處理的速度遠高於 30FPS，所以未來如果更換更好的 camera 可以再加快模型的判斷速度。

#### 三、 設計一個高畫質解析電路供 Camera 輸入

目前我們使用的 camera 的輸入畫質為 640\*480，如果換成高畫質的輸入可換得更詳細的特徵，進一步提升模型的準確度。