

# 視覺辨識模型在 FPGA 上實作

## Implementation of computer vision model on FPGA

指導教授：鄭桂忠

組別：A119

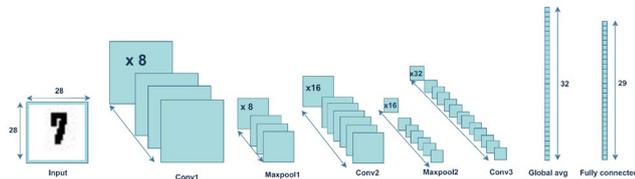
組員：林鈺期、謝郁楷

### Abstract

隨著科技進步，電腦運算能力及機器學習演算法的發展，機器視覺的技術如 image classification 或 object detection 也受到重視。這些技術主要基於 convolution neural network (CNN)，由於龐大的運算量，我們可以針對這個特性，設計一個平行運算的硬體加速它，設計這樣的硬體，我們主要是利用 FPGA 來實現，除此之外，由於低延遲以及低功耗的特性，使得這項技術都更適合被運行在 FPGA 上。然而雖然有上述優點，但由於 FPGA 上有限的儲存空間與運算資源，這也使得運行這項技術於 FPGA 上是具有挑戰性的。

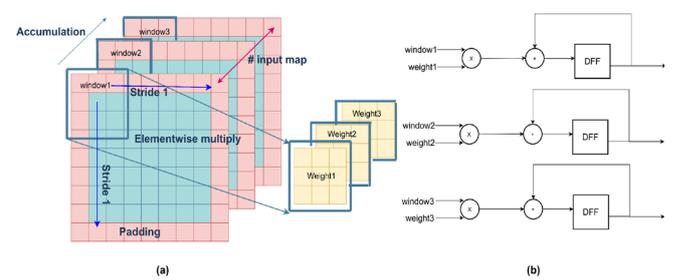
我們的專題為建立 FPGA based 的 CNN 網路的視覺辨識加速器，在這專題我們使用 Xilinx PYNQ-Z2 開發板，以及 webcam。一開始我們首先了解神經網路的架構及原理，之後使用 Tensorflow-Keras 來訓練出網路，並熟悉開發板上 AXI 資料傳輸介面，以及了解一些如何盡量重複利用硬體及減少存取記憶體的方式，最後利用 Verilog 硬體描述語言來將訓練出來的網路實裝到 FPGA 上，實現簡單的視覺辨識神經網路。

### CNN 網路架構

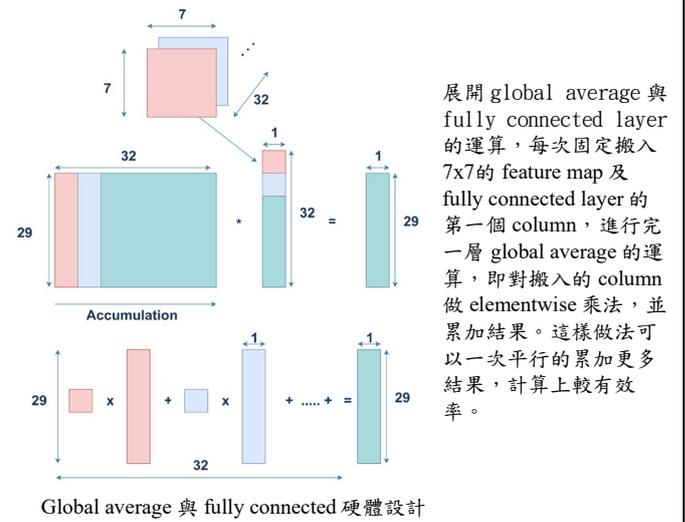


網路架構圖

Input : 28 x 28 x 1 灰階數字、符號  
3 Convolution layer 2、2 Maxpooling layer  
Global average pooling  
Fully connected layer  
Output : 29 classes probability

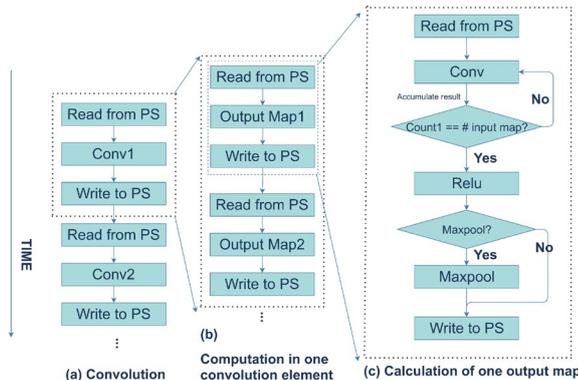


Convolution 硬體設計



展開 global average 與 fully connected layer 的運算，每次固定搬入 7x7 的 feature map 及 fully connected layer 的第一個 column，進行完一層 global average 的運算，即對搬入的 column 做 elementwise 乘法，並累加結果。這樣做法可以一次平行的累加更多結果，計算上較有效率。

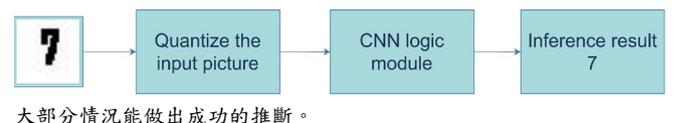
### 硬體設計



硬體運算流程圖

因為一次能儲存的 register 有限，因此必須有頻繁搬運資料的情況。若能減少資料搬動，可以提升效率。整個過程拆成三個 convolution，如圖 (a)。在每個 convolution element 內要運算很多層 output feature map，如圖 (b)。每層 output feature map 的運算如圖 (c)，運用累加的方式減少資料搬出，並重複使用硬體。

### 結果



### 結論

這次專題主要是實作在 FPGA 上實現視覺辨識的網路，我們設計在 CNN 運算 module 中採用的運算及資料搬運方式，最終能夠實現在 FPGA 平台上辨識出輸入圖片的類別。希望未來能在此基礎上實現建立更大型、更有效率的 FPGA 硬體加速器。

### 參考資料

[1] CHEN C, CHAI Z L, XIA J. Design and implementation of YOLOv2 accelerator based on Zynq7000 FPGA heterogeneous platform. Journal of Frontiers of Computer Science and Technology 2019.