適用動畫超解析之卷積神經網路處理器設計

Convolutional Neural Network Processor for Animation Super Resolution

組別:A49　組員:張嘉祐、林俊曄、陳昭霖

指導教授:黃朝宗 教授

摘要

在這個科技發達的世代，很多人使用手機或平板看動畫，隨時隨地都可以觀賞自己喜歡的動畫，但在網路不好的狀況下，傳輸速率變差，導致畫質下降，降低了我們的觀看品質。如果先行下載高畫質的影片至手機內，不僅要等待下載的時間而且會占掉許多寶貴的記憶體空間。因此我們想訓練出一個針對動畫風格的超解析之 CNN 網路，讓畫質較差的圖片經過這個網路後解析度能提高，變得更加清晰，並將這個網路以硬體實作出來，使 Throughput 能達到動畫等級的規格(每秒輸出24張圖)，完成一個 CNN 硬體加速器。最終目標為將此硬體應用於手機或平板這種方便攜帶的電子產品上，讓我們以少量的記憶體空間存放 weight、bias、運算過程的 activation，便能觀賞高畫質的動畫。不僅可以節省記憶體空間，也能在網路傳輸速率不佳的情況下保有高畫質的觀賞體驗。

在這個專題研究中，我們決定訓練一個針對海賊王畫風的 SR-CNN model，並實作在硬體上。我們的研究計畫是一個完整的系統設計，涵蓋軟體模擬、硬體模擬與合成、APR 三大部分。軟體部分包含 Hyperparameters 的分析與選擇，Data 的 Quantization 和 Dynamic fixed point 等提取策略，目的都是為了讓原本在軟體上的架構移駕到硬體上時表現不會降低太多；硬體部分包含架構的設計、模擬驗證及優化；APR 部分包含利用 Synopsys Design Compiler 合成電路、在 TSMC 40nm 製程的環境下使用 IC Compiler 製作 APR Layout，最後分析合成電路及 APR Layout 的 Timing、Area、Power。

我們最終 APR 結果的 Timing 是 5.4ns，Area 是 936079μm2，Dynamic Power 是 50.96mW，Leakage Power 是 4.46mW，輸出一張高畫質圖片的總 Cycle1 數為 10,715,029 個，由此可算出 throughput 為每秒 17.28 張。輸出圖片與用於訓練卷積神經網路的高清圖片比較後的 PSNR 平均為 29.11dB，以人眼觀測輸入與輸出的圖片畫質有顯著的提升。

一、研究內容

我們的研究計畫是一個完整的系統設計。首先利用 Pytorch 訓練出針對海賊王動畫畫風的超解析之 CNN 網路，接著以 Verilog 設計這個網路的數位電路，最後使用 TSMC 40nm 的製程完成 APR 分析。現在的日本動畫，多數以每秒24幀來計算，也就是每秒24張圖片。因此我們的目標 throughput 為每秒輸出24張高解析度的圖，確保我們的設計能實際應用於動畫上。

在 Model 中我們使用 ResNet 架構，從較新的動畫集數中隨機收集約1000張的高清圖片，經過模糊處理後形成1000組 Data pairs(Low Resolution, High Resolution)作為 Data Set 去訓練 Anime-ResNet Model，並以 PSNR 作為 Performance 的指標，最後從所有訓練出的 Model 中選取一個最適合的 Model 實作在硬體上。

在數位電路設計中，我們將訓練好的參數(Weight、Bias)存進 SRAM 裡，並用另外兩個 SRAM 交互存放運算過程中的 Activation。輸入為一張大小為320x180 pixels 的海賊王圖片，輸出為一張大小1280x720 pixels 的高解析度海賊王圖片。最後我們使用 TSMC 40nm 的製程來完成 APR，並對結果的 Timing、Area、Power 進行分析與討論。

二、Implementaion

I. Anime-ResNet Model

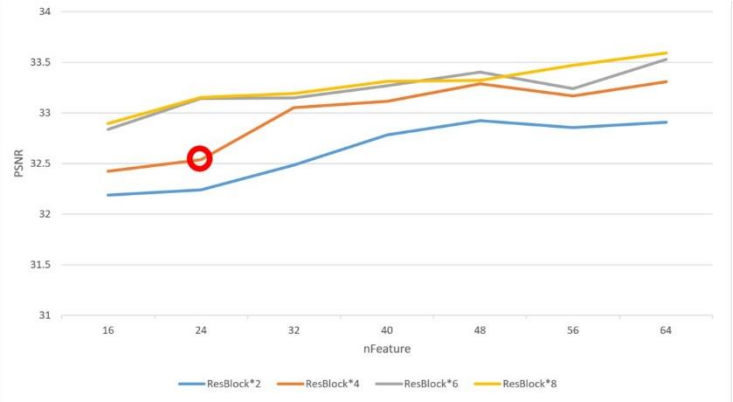


圖 1　不同規格 model 的比較(紅圈是我們的 model)

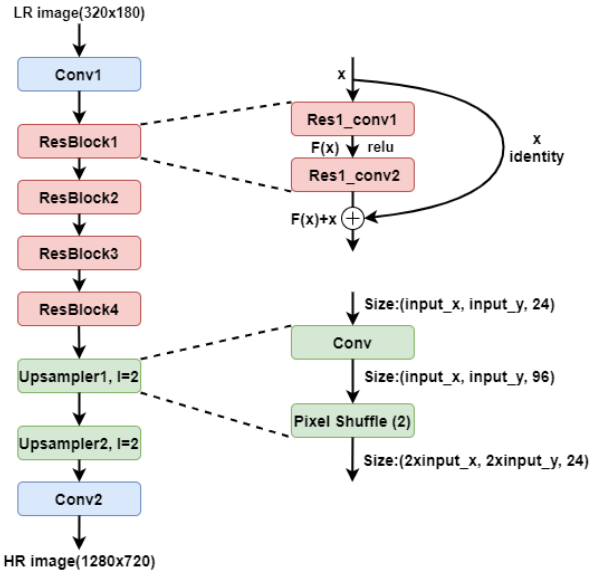


圖 2　Anime-ResNet model Structure

我們分別訓練圖 1 這些 Model，以 20 張不在 dataset 內的圖片測試，平均後得出其相對應的 PSNR，並衡量 Performance 與該 Model 實作在硬體上所需的記憶體空間。我們最終決定採用 ResBlock*4 和 nFearture=24 的架構，如表 1 所示。

Model 的架構如圖 2 所示，包含四個 ResBlock unit 和兩個 upsampler，因此 output 的圖片大小為 input 的 16 倍。

Parameter	nFeat	nResblock	nEpochs
Value	24	4	200

表 1　Anime-ResNet Model Specification

利用 Dynamic Fixed Point 可以將硬體的資源最大化。在硬體上不適合用 32bits 的浮點數運算，因此我們分析每層 Parameters 跟 Activation 的分布情形，再擷取最佳範圍，將每個 bit 的效用發揮到最大。例如圖 3 所示，左圖是還未經過 Dynamic Fixed Point 調整的 upsampler 2 activation 分布圖，可觀察到有 saturation 的現象，因此我們將小數部分的 1 個 bit 分給整數部分，以完整表達所有整數部分。

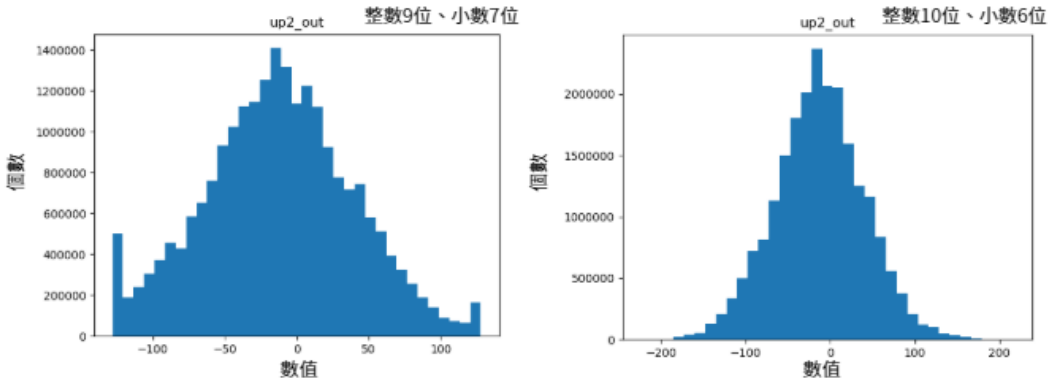


圖 3　Dynamic Fixed Point Operation

II. 硬體架構設計

在數位電路設計中，為了使 Throughput 能達到動畫等級的規格(每秒輸出 24 張圖)，我們使用 864 個乘法器，一個 cycle 可以輸出四個值存進 SRAM。運算過程我們使用兩組相同規格的 SRAM(A、B)交互存放 activation。表 2 為硬體規格與 SRAM 規格。

Input	320x180 image
Output	1280x720 image
乘法器數量	864個
SRAM for activation	45MB x2個 (SRAM_A&SRAM_B)
SRAM for weight	88KB x1個
SRAM for bias	0.4KB x1個

表 2　Hardware Specification

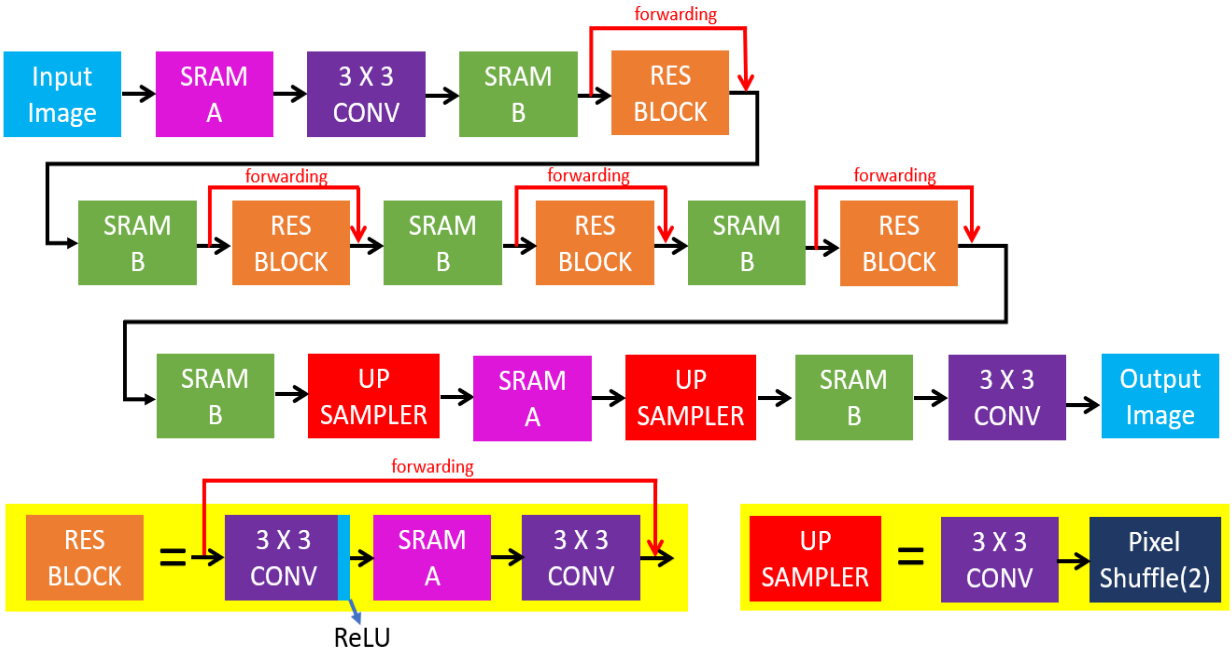


圖 4　Data Flow

在 Resblock Unit 中有一個比較特別的部分是 Forwarding，也就是在 Resblock unit 中的第二層 3x3 Convolution 運算完後要加上第一層 3x3 Convolution 的 Input。由於我們使用兩組 SRAM(Group A、Group B)交互存取資料，第二層 3x3 Convolution 運算完後要寫入 SRAM B，而第一層的 Input 也是存在 SRAM B，我們使用先讀後寫的方法，利用 Pipeline 錯開讀值與寫值的時間，先將 Forwarding 的值從 SRAM B 讀出來，處理完加法後再寫回去 SRAM B 的同一位置。如此雖然會有多一個加法運算的 Overhead，但就不必先存進 SRAM 再讀回來進行 Forwarding 的加法運算，可以節省大量的 Cycle，以及不用再多開一組 SRAM 去存值。

三、成果與分析

結果分析我們分成 20bits 的 Activation 以及 16bits 的 Activation 作為探討，20bits 的版本目前已驗證 Gate Level Simulation 且完成 APR，而 16bits 版本目前只有完成到 HDL Simulation 驗證，所以沒有完整的 APR Layout 無法與合成的結果比較，因此 16bits 的部分只討論合成後的結果。

Activation: 16bits	Synthesis
Timing	3.8ns
Area	$695695\mu m^2$
Power(dynamic):	$48.59mW$
Power(leakage):	$3.11 \times 10^3 \mu W$
Total Cycles	10,715,029 個
Throughput	24.56 image/sec

表 3　Specification of 16 bits Activation

Activation: 20bits	Synthesis	APR (Utilization 0.5)
Timing	3.8ns	5.4ns
Area	$844568\mu m^2$	$936079\mu m^2$
Power(dynamic):	$56.046mW$	$50.96mW$
Power(leakage):	$3.57 \times 10^3 \mu W$	$4.46 \times 10^3 \mu W$
Total Cycles	10,715,029 個	10,715,029 個
Throughput	24.56 image/sec	17.28 image/sec

表 4　Specification of 20 bits Activation

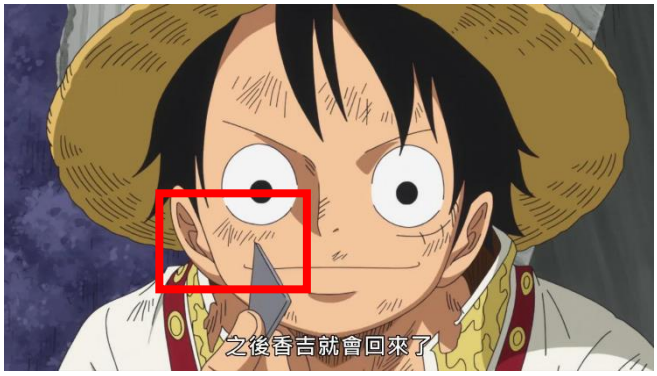


Image01 from the latest animation

HR (PSNR)

Anime-ResNet (Pytorch)

(34.736dB)

Gaussian Blur

Anime-ResNet (Hardware)

(30.254dB)

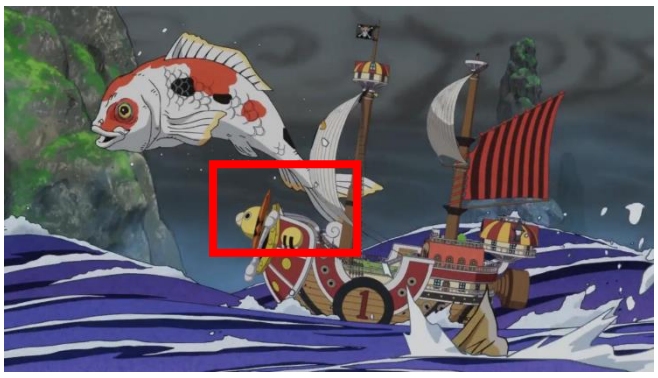
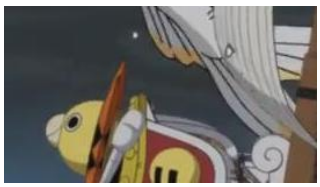
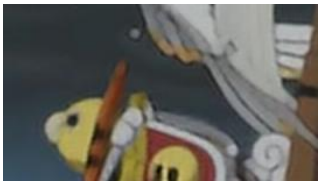


Image04 from the latest animation

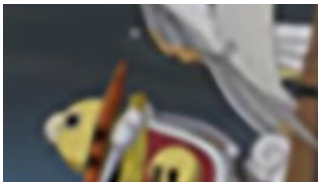


HR (PSNR)

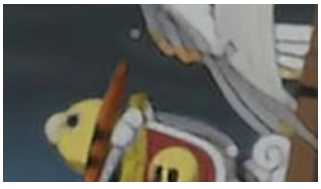


Anime-ResNet (Pytorch)

(33.970dB)



Gaussian Blur



Anime-ResNet (Hardware)

(31.425dB)

圖 5　軟體硬體成果比較

四、心得

張嘉祐：

在選擇這次專題研究主題的機會下，我踏入了數位設計的世界，體會到整個 Digital Design Flow 中的每一個步驟都是非常嚴謹的。從軟體演算法的選擇、硬體的架構設計與合成、到 APR 的策略，每個環節間都有密切的關聯性，團隊間的溝通與合作至關重要。在團隊中我負責硬體的部分，包含整體架構設計、記憶體大小的選擇、RTL 模擬及優化。過程中雖然遇到許多瓶頸，像是硬體與軟體的 Output 一直有些許誤差、一開始的 I/O pin 數過多使 APR 在做 Floor plan 時有困難，但經過一次又一次的溝通及調整，最後都順利跨越層層阻礙。很感謝實驗室的學長姐們在我們遇到困難時不厭其煩地協助我們解決許多問題，也謝謝教授在每次 Meeting 時給予許多建議及方向，讓原本沒有什麼想法、猶豫不決的我們慢慢有了明確的目標，一步步朝目標邁進，成功完成專題研究。經過一年的磨練，我對數位設計領域有了更深一層的理解，也培養了許多相關的能力與知識，更重要的是學到如何在團隊中扮演好自己的角色，與組員們互相分工合作，過程雖然辛苦卻很值得。

林俊曄：

在這次專題，我負責軟體的部分，主要是利用 Pytorch 來訓練和優化 Model，並找出適合實作在硬體上的參數提取策略。透過這次的經驗，我深刻體會到軟硬體架構的差異，還有最後實際 layout 時需要特別注意的重點。在軟體上，我不用特別考慮 Paramter 的存取方式，全都採用 Float point 32bits 的資料型態，在 Anime-ResNet 上能達到最佳的效果；但為了與硬體整合，多了資料 bits 數上的限制，需要找出不同 data 最適合的區間，才會讓損失的 PSNR 降到最低。此外，軟體與 Layout 的關係也很重要，當初我們版本一的規格是 Activation 20 bits/ Parameters 8 bits ，但是在這樣的架構下，I/O pin 數目和整體面積過大會不符合經濟效益，因此在第二版本 Activation 16bits/ Parameter 8 bits 的情形下，能大幅降低上述的問題，並且也保有令人滿意的圖片品質。我從這次的專題中學習到如何與其他成員和學長姐們合作與討論，我以往會因為害怕自己的問題沒有建設性或太愚蠢就害怕向他人詢問，但隨著多次與教授和學長的 Meeting 後，我發現如果自己有哪裡不清楚就不要害怕提出自己的問題，因為比起自己埋頭苦幹浪費許多時間，直接向有經驗的前輩詢問真的會學習到許多意想不到的收穫。

陳昭霖：

透過這一次的專題實作，我負責最後 APR Layout 的部分，讓我深切的體會到硬體設計之間上下游的關係，從上學期修習積體電路設計實驗學習使用 32nm 虛擬製程，到後來將我們使用實際 TSMC 40nm 製程，製作能實際去台積電下線的 Chip，其實相當的令人興奮。從一開始軟體架構的選擇，到硬體設計過程的優化，都會有相當程度影響到最後的合成電路及 APR Layout，我大部分的時間在熟悉 Design Compiler、IC

Compiler、EDA Cloud 的操作，以及測試各種不同的合成策略，包含合成的 Timing Constraint、Floor plan、Power 規劃等，而我們的硬體設計面積約 800000μm^2，是一個不小的設計，光從 Design Compiler 所要合成 Netlist 就需要等待 4 到 8 小時，之後的 APR Layout 更是要再花 5 小時的時間才能夠看到最後的成果是否符合我們的要求，而每一次的測試都需要花費半天到一天的時間，其實相當的耗時耗神。但是幸虧這一次的專題實作，能讓我們一窺數位電路設計的好玩之處，看到我們的成果能有實際的成品，就額外的有成就感，期待我們之後有機會繼續朝這個領域的技術及研究發展，努力增進自己的實力。