Deep Learning – Assigment II

數據一 RE6121011 徐仁瓏

壹、 介紹

本報告旨在探討深度學習的兩個任務，並提出相應的解決方案。首先，我們將簡要概述這兩個任務的目標和要求。隨後，我們將介紹一個重要的數據集，即 ImageNet-mini 數據集，這一個數據集將用於訓練和測試我們的模型。最後，我們將強調在評估模型性能時驗證集和測試集的重要性，這有助於確保模型的泛化能力和準確性。

貳、 數據集

ImageNet-mini 數據集是一個小版本的 ImageNet 數據集，其目的是在保留較大數據集的基本特性的同時減少計算複雜度。該數據集包含了訓練、驗證和測試集，這些集合將用於訓練和評估我們的模型。透過訓練集，模型可以學習數據的特徵和模式，而驗證集則用於調節超參數以改善模型的性能。最後，測試集用於評估模型在未見過數據上的表現，這有助於確定模型的泛化能力和準確性。因此，驗證集和測試集的性能評估對於確保模型的可靠性和有效性至關重要。

參、 任務一：

設計一個針對可變輸入通道的卷積模塊

一、 任務描述

卷積神經網絡（CNN）在計算機視覺任務中取得了巨大成功，但傳統的 CNN 模型在處理可變輸入通道的情況下存在一定的限制。為了解決這個問題，我們的目標是設計一個特殊的卷積模塊，該模塊能夠處理任意數量的輸入通道，並且具有空間尺寸不變性。我們將詳細描述設計原則、實現細節以及實驗設計。

二、 設計原則

我們採用了動態卷積的概念，設計了一個動態卷積模塊，該模塊能夠根據輸入通道的數量動態調整其權重。我們參考了吳等人在 CVPR 2020 年發表的論文《動態卷積：關注卷積核》。

動態卷積神經網絡（DY-CNN）的目標是在網絡性能和計算負擔之間提供更好的權衡，它不會增加網絡的深度或寬度，而是通過注意力聚合多個卷積核來增加模型能力。對於不同的輸入圖像，這些內核的組裝方式不同，從動態卷積得名。概念流程如圖 1 所示。

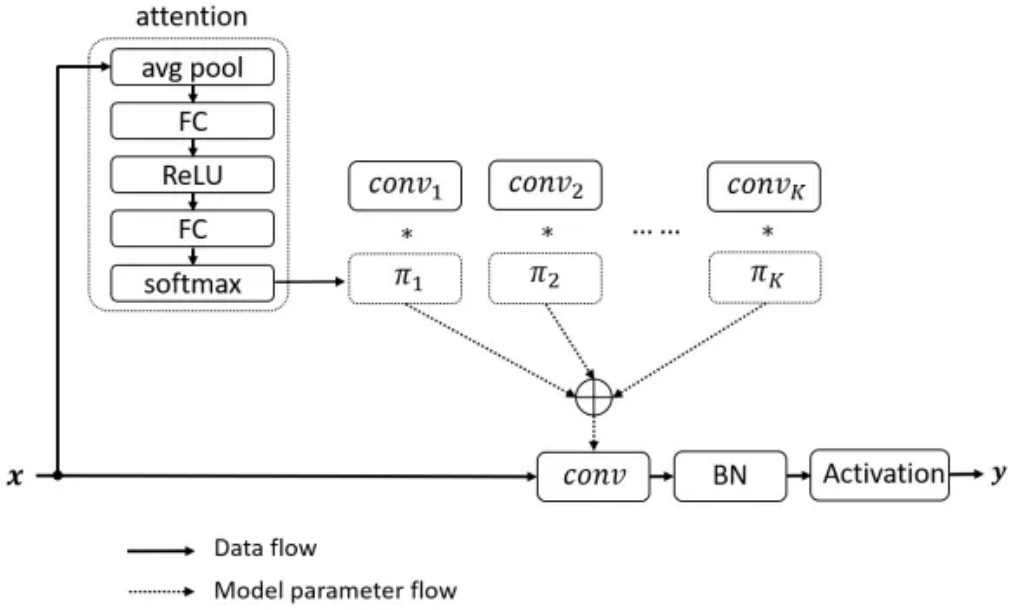

圖 1 動態卷積網路之架構

首先，動態卷積中的注意力機制是通過擠壓和激發（squeeze-and-excitation）來計算卷積核的注意力權重π_K。首先，將輸入的全局空間信息進行全局平均池化，這一步稱為擠壓，旨在匯集整個輸入特徵圖的全局信息。接著，這些壓縮後的全局特徵通過兩個全連接層，其中在這兩層之間加入了一個 ReLU 激活函數，以引入非線性變換。最終，通過 softmax 函數生成歸一化的 K 個卷積核的注意力權重。這些注意力權重反映了每個卷積核對於當前輸入的相對重要性。值得注意的是，第一個全連接層將神經元減少到 K＝4 個，以降低計算成本，同時保留必要的全局信息。

接著，動態卷積中包含 K 個卷積核，這些卷積核共享相同的內核大小以及相同的輸入和輸出維度。通過使用注意力權重π_K，將這些卷積核進行聚合。隨後，遵循卷積神經網絡（CNN）中的經典設計原則。在完成卷積核的聚合後，動態卷積層進一步使用批量歸一化和激活函數（例如 ReLU）來增強模型的性能和穩定性，從而構建出功能強大的動態卷積層。

三、實現細節

我們實現了一個基於 VGG19 網絡的深度學習模型，該模型使用了動態卷積層來增強特徵表示能力。動態卷積層通過引入注意力機制，根據輸入圖像的不同特徵動態地調整卷積核的權重。

動態卷積層使用多個卷積核，這些卷積核共享相同的內核大小和輸入/輸出維度。動態卷積層通過計算注意力權重來動態地聚合這些卷積核。注意力機制使用擠壓和激發(squeeze-and-excitation)的方法來計算卷積核的注意力權重。首先，對輸入特徵圖進行全局平均池化，得到全局特徵向量。接著，這些全局特徵向量通過兩個全連接層進行變換，中間使用 ReLU 激活函數。最後，通過 softmax 函數生成歸一化的 K 個卷積核的注意力權重。動態卷積層使用計算出的注意力權重來動態地聚合多個卷積核的權重，從而得到最終的卷積結果。具體來說，對於每一個輸入樣本，根據注意力權重來線性組合 K 個卷積核的權重，生成一個特定於該樣本的卷積核，並用這個卷積核對輸入進行卷積操作。

最後，我們對比了傳統 VGG19 與融入動態卷積 VGG19 的模型參數與計算成本，具體比較如表 1 所示。

表 1 比較模型的計算成本

MODEL	FLOPS	#PARAMS
VGG19	25,578,677,248	21,289,842
VGG19 + dynamic conv	25,672,863,320	81,885,642

透過融入動態卷積，雖然可能會增加額外的計算成本和參數數量，但可以顯著減少為不同通道數量訓練和存儲模型的需求，讓我們可以直接在不同通道下進行推論，而不需要重新訓練其他通道數的模型架構。

四、實驗結果

我們使用 ImageNet-mini 數據集進行模型的訓練，訓練期間的參數設置皆相同。在測試期間，我們使用具有不同通道組合（例如 RGB、RG、GB、R、G、B 等）的圖像分別對傳統 VGG19 與融入動態卷積 VGG19 的模型進行測試，以評估兩個模型在處理不同通道組合時的性能。

為了讓動態卷積模組的模型與天真模型進行公平的比較，因為傳統 VGG19 模型只能處理三個通道的輸入圖像，因此我們對單通道和雙通道的輸入進行以下處理：

- 單通道（R, G, B）輸入： 將單個通道複製兩次，生成三個通道數的圖像作為輸入。
- 雙通道（RG, GB）輸入： 將兩個通道的像素取平均值，生成第三個通道數，從而生成三個通道數的圖像作為輸入。

這樣處理後，我們可以將不同通道組合的圖像作為三個通道數的輸入圖像，並輸入到模型中進行測試。從下表 2 到下表 7 中，我們比較了融入動態卷積 VGG19 與傳統 VGG19 模型在 ImageNet-mini 數據測試集上的性能，並從以下幾個方面進行評估：準確性（Accuracy）、精度（Precision）、召回率（Recall）、F1 得分（F1-score）。

表 2 比較模型在 RGB 通道的測試性能

Model	Accuracy	Precision	Recall	F1-score
VGG19	0.2911	0.2776	0.2911	0.2691
VGG19+dynamic conv	0.2933	0.2983	0.2933	0.2762

表 3 比較模型在 RG 通道的測試性能

Model	Accuracy	Precision	Recall	F1-score
VGG19	0.3111	0.3159	0.3111	0.2871
VGG19+dynamic conv	0.3333	0.3038	0.3333	0.3034

表 4 比較模型在 GB 通道的測試性能

Model	Accuracy	Precision	Recall	F1-score
VGG19	0.3133	0.3142	0.3133	0.2948
VGG19+dynamic conv	0.3489	0.3302	0.3489	0.3131

表 5 比較模型在 R 通道的測試性能

Model	Accuracy	Precision	Recall	F1-score
VGG19	0.32	0.3368	0.32	0.2936
VGG19+dynamic conv	0.3578	0.3336	0.3578	0.3205

表 6 比較模型在 G 通道的測試性能

Model	Accuracy	Precision	Recall	F1-score
VGG19	0.3422	0.3721	0.3422	0.3216
VGG19+dynamic conv	0.3956	0.3937	0.3956	0.3608

表 7 比較模型在 B 通道的測試性能

Model	Accuracy	Precision	Recall	F1-score
VGG19	0.3133	0.3187	0.3133	0.2933
VGG19+dynamic conv	0.3711	0.3695	0.3711	0.3412

從以上結果可以看出，動態卷積模組的模型在處理可變輸入通道時具有顯著的性能優勢。儘管增加了計算成本和參數數量，但性能提升使得這種額外的成本是值得的。尤其是在通道數越少時，融入動態卷積 VGG19 的各項指標都比傳統 VGG19 的性能要好。我們的模型在不同通道組合下均表現出色，顯示了其在處理可變輸入通道數據方面的強大能力。這些結果證明了動態卷積模組的有效性，為未來在計算機視覺領域中的應用提供了有力的支持。

五、消融實驗

為了進一步驗證動態卷積模組的有效性，我們針對其注意機制中第二層全連接層的輸出通道數 K 進行了消融實驗。我們設置了四個不同的輸出通道數 K 為 3、4、5 和 6，並在相同的訓練條件下進行實驗，最終比較驗證集上的準確性。

表 8 通道數 K 的設置對於驗證集上準確性的實驗結果

K	3	4	5	6
Accuracy	0.5632	0.5644	0.5617	0.5589

從實驗結果可以看出，不同的輸出通道數對模型的性能有一定影響。當 K 增加時，準確性並沒有顯著提升，反而在 K＝4 時達到最佳值。這表明過高的輸出通道數可能會引入冗餘信息，從而影響模型性能。通過消融實驗，我們確認了在動態卷積模組中，設定第二層全連接層的輸出通道數 K 為 4 是最佳選擇。這為我們的最終模型設計提供了重要的依據，並顯示了消融實驗在優化模型配置中的重要性。

六、結論與展望

在本研究中，我們成功設計並實現了一個針對可變輸入通道的動態卷積模塊，並在 ImageNet-mini 數據集上進行了廣泛的實驗。實驗結果顯示，與傳統的 VGG19 模型相比，融入動態卷積的 VGG19 模型在處理不同輸入通道組合時均表現出顯著的性能優勢。我們的動態卷積模塊通過注意力機制動態調整卷積核的權重，有效地增強了特徵表示能力。具體來說，我們的動態卷積模型在多個通道組合下均表現優異，特別是在單通道和雙通道輸入的情況下，顯著超越了傳統模型。此外，通過消融實驗，我們確定了最佳的注意力機制設置，進一步提升了模型性能。儘管動態卷積模塊增加了部分計算成本和參數數量，但其性能提升使得這些額外成本是值得的。

未來，我們計劃從多個方面進一步研究和改進動態卷積模塊。首先，儘管動態卷積模塊展示了優越的性能，但其增加的計算成本仍需進一步優化。我們將探索更高效的注意力機制和權重生成方法，以降低模型的計算複雜度。同時，我們的研究主要基於 VGG19 模型，未來可以將動態卷積模塊應用於其他深度學習模型（如 ResNet、DenseNet 等），以驗證其在不同架構下的通用性和效果。

肆、 任務二：

設計一個四層網絡進行圖像分類

一、 任務描述

設計一個 2-4 層的 CNN、Transformer 或 RNN 網絡，該網絡在 ImageNet-mini 上可以達到 ResNet34 約 90% 的性能（即性能損失不超過 10%）。參數數量或 FLOPS 沒有任何限制，但輸入和輸出層的最大數量限制在 4-6 之間。

二、 設計原則

我們的設計原則是採用 RRDB（Residual-in-Residual Dense Block）以增加感受野並增強特徵提取，同時引入 Attention 機制以提高全局特徵的捕捉能力。這兩項技術的結合旨在構建一個高效且精簡的網絡，在有限層數的條件下達到 ResNet34 90%的性能。

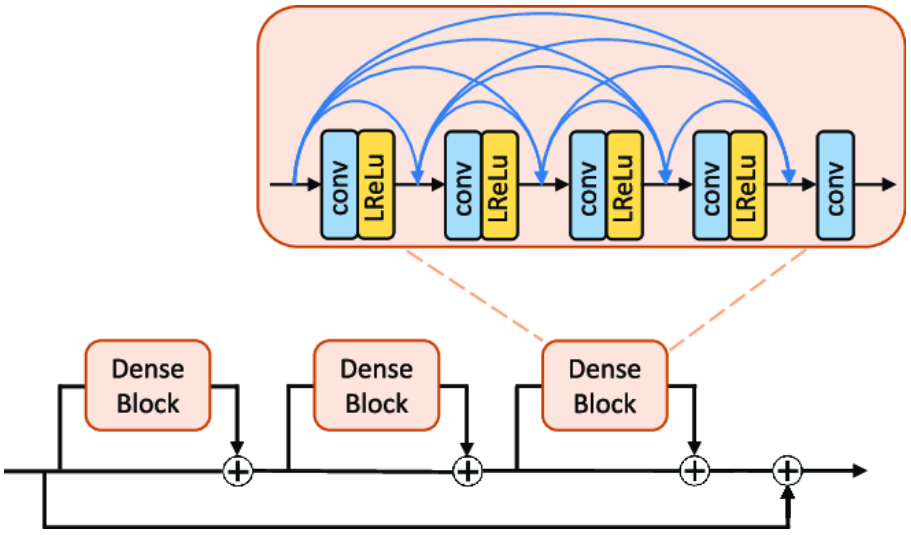

圖 2 RRDB 架構

RRDB 的概念源自 Wang 等人在 ECCV 2018 上發表的論文“ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks”。RRDB 結合了殘差塊和密集塊的優點，形成了一種高效的特徵重用和增強方法。具體而言，如圖 2 所示，RRDB 在每個密集塊內引入了殘差連接，這樣可以有效地解決深層網絡中的梯度消失問題，並且增強了特徵的傳遞和重用能力。通過多層次的特徵重用，

RRDB 能夠顯著增加網絡的感受野，使得網絡能夠更深入地提取圖像中的重要特徵。這種設計在圖像超分辨率任務中已經證明了其優越性，我們將其應用於我們的網絡結構中，以期在圖像分類任務中同樣發揮其強大的特徵提取能力。

在我們的設計中，除了 RRDB 模塊，我們還引入了 Attention 機制，特別是自注意力機制，以進一步提高全局特徵的捕捉能力。自注意力機制是模仿人類注意力的一種方法，它通過對輸入特徵圖中的每個位置計算其與其他位置的相關性，來動態調整特徵的重要性。這種機制可以使網絡更好地理解圖像中的全局上下文信息，從而提高分類性能。自注意力機制不僅能夠捕捉長距離的依賴關係，還能有效地聚焦於圖像中的關鍵區域，使得網絡能夠對整個圖像進行更全面的分析和理解，架構如圖 3 所示。

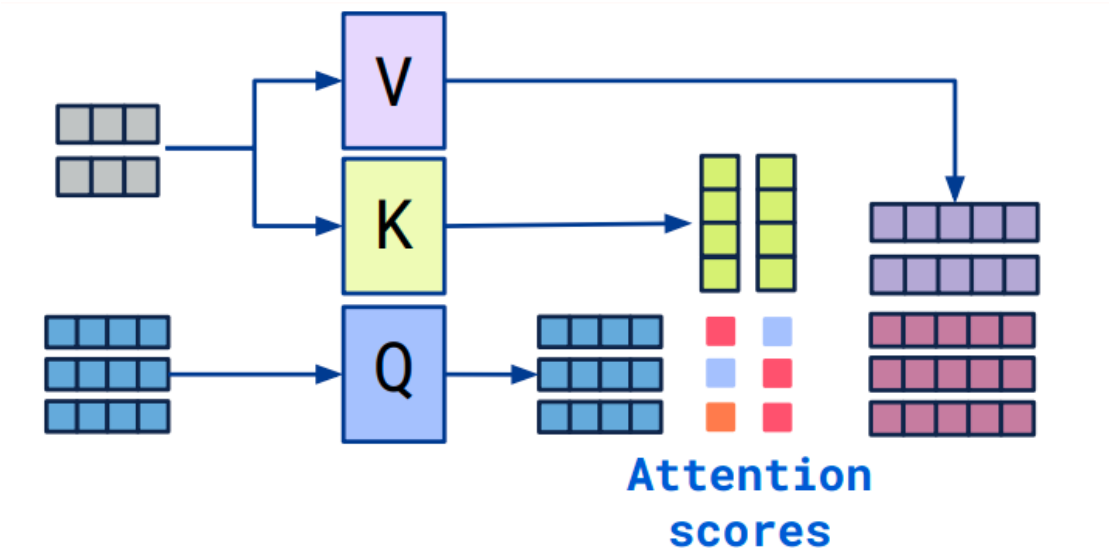

圖 3 Attention 機制

三、實現細節

網絡架構是一個四層的深度學習模型，設計目的是在有限層數的情況下達到高效的圖像分類性能。該網絡包含兩個卷積層，中間並聯了一個 RRDB 模塊和一個自注意力機制，最後通過全連接層輸出分類結果。以下是詳細的實現細節。

首先，網絡的第一層是一個卷積層。這一層接收輸入圖像，其通道數為 3（對應 RGB 圖像）。卷積核的數量設為 64，卷積核大小為 3x3，步長為 1，並且使用了填充大小為 1 的設置，以確保輸出特徵圖的空間維度與輸入保持一致。經過這一層後，特徵圖的通道數從 3 增加到 64。隨後，通過批量正規化（Batch

Normalization)，進一步提高訓練過程中的穩定性。激活函數選用了 ReLU，以引入非線性特性。第一層最後是一個最大池化層（MaxPooling)，池化窗口大小為 2x2，步長為 2，用於下采樣特徵圖，減少計算量和參數數量。

接下來的部分是網絡的核心，包括 RRDB 模塊和自注意力機制。首先是 RRDB 模塊，其設計目的是通過殘差內殘差密集塊來增強特徵提取能力。RRDB 的輸入通道和輸出通道都為 64，這意味著 RRDB 模塊處理的特徵圖通道數保持不變。RRDB 模塊能夠有效地重用特徵，增加感受野，使網絡在捕捉圖像細節方面更加出色。RRDB 包含了一個密集塊（DenseBlock)，每個密集塊包含三層卷積層。第一層卷積接收的輸入通道數為 64，輸出通道數為 32（growth rate)。第二層卷積接收的輸入通道數為 96（來自原始輸入和第一層的輸出)，輸出通道數為 32。第三層卷積接收的輸入通道數為 128（來自原始輸入、第一層和第二層的輸出)，並輸出 64 個通道數以匹配原始輸入大小。每層卷積之後都應用 ReLU 激活函數，並使用殘差連接來進一步增強特徵表現。RRDB 模塊的輸出是通過將所有卷積層的輸出進行連接，乘以 0.2 然後加上原始輸入，這樣可以更好地保留原始信息。

與 RRDB 並聯的是自注意力機制。自注意力機制也接收和輸出相同通道數的特徵圖，即 64 通道。自注意力機制的引入使網絡能夠更好地捕捉全局特徵，通過計算特徵圖中不同位置之間的相關性，動態調整特徵的重要性。這有助於網絡在處理複雜圖像時更加精確。具體來說，自注意力機制通過三個 1x1 的卷積層來生成查詢（query)、鍵（key）和數值（value）映射。首先，輸入特徵圖通過查詢和鍵卷積層，分別生成具有 C/8 通道的特徵圖，然後這些特徵圖被展平成矩陣，計算其內積以生成注意力權重矩陣。接下來，輸入特徵圖通過數值卷積層生成具有 C 通道的特徵圖，同樣展平成矩陣，然後與注意力權重矩陣進行矩陣乘法以得到最終的注意力輸出。這個輸出再經過縮放係數 γ 進行縮放，最後與原始輸入進行加和，形成自注意力機制的最終輸出。RRDB 和自注意力機制的輸出被融合在一起，通過簡單的加法操作將兩者的輸出相加。這一操作將來自兩個不同特徵提取路徑的信息合併，使網絡能夠充分利用這兩個模塊的優勢。

網絡的第三層是另一個卷積層。這一層的輸入通道數為 64，輸出通道數設為 128。卷積核大小同樣為 3x3，步長為 1，填充大小為 1。通過這一層，特徵圖的通道數從 64 增加到 128。隨後的處理步驟包括批量正規化和 ReLU 激活函數，用於正規化和非線性轉換。這一層最後也包含一個最大池化層，池化窗口大小為

2x2，步長為 2，進一步下采樣特徵圖。

在這之後，網絡進入全局平均池化層（Global Average Pooling）。這一層將特徵圖的空間維度壓縮到 1x1，僅保留每個通道的平均值，這樣可以減少參數數量並保持最重要的特徵信息。最後，網絡的輸出層是一個全連接層（Fully Connected Layer），將 128 個輸出通道映射到分類節點。假設輸入圖像的大小是 64x64，經過前面的層後，特徵圖的尺寸將變為 1x1，具有 128 個通道。全連接層將這 128 個特徵映射到 50 個輸出節點，對應於 50 個分類類別。

總結來說，這個網絡架構通過兩層卷積層、RRDB 模塊、自注意力機制和全連接層的組合，實現了有效的特徵提取和分類。每一層的設計都經過精心考慮，以平衡網絡的複雜性和性能，達到高效的圖像分類效果。

四、實驗結果

在這部分，我們詳細介紹了所設計網絡和 ResNet34 在 ImageNet-mini 數據集上的性能評估結果，包括準確性、計算成本，以及這些結果的定量比較。實驗結果展示了我們所設計的網絡在有限資源條件下的有效性和潛力。

首先，我們比較了兩個模型的計算成本，具體分析了 FLOPS 和參數數量，如表 8 所示。ResNet34 的 FLOPS 為 4,809,609,216，參數數量為 21,797,672；而我們所設計的 AdvanceTwoLayerCNN 的 FLOPS 為 7,319,591,296，參數數量為 447,490。雖然我們的模型在 FLOPS 方面高於 ResNet34，但其參數數量大大減少，僅為 447,490，這意味著我們的模型在存儲和部署方面具有更大的優勢。此外，這些數據顯示我們的模型在保證較低參數數量的同時，仍能達到較高的運算能力，這在某些資源受限的應用場景中具有重要意義。

接下來，我們進行了準確性的評估，結果如表 9 所示。所設計的 AdvanceTwoLayerCNN 在 ImageNet-mini 數據集上的分類準確性為 43.11%，而 ResNet34 在同樣數據集上的分類準確性為 47.78%。雖然我們的模型在準確性上略低於 ResNet34，但其性能損失在可接受範圍內，即不超過 10%的差距。此外，我們還計算了兩個模型的精確度、召回率和 F1 分數。ResNet34 的精確度、召回率和 F1 分數分別為 53.92%、47.78%和 47.14%；而我們的 AdvanceTwoLayerCNN 的這些指標分別為 45.80%、43.11%和 42.61%。這些結果表明，雖然我們的模型在一些評估指標上略有劣勢，但其整體表現仍然接近於 ResNet34。

表 8 比較模型的計算成本

MODEL	FLOPS	#PARAMS
ResNet34	4,809,609,216	21,797,672
AdvanceTwoLayerCNN(Ours)	7,319,591,296	447,490

表 9 比較模型的測試性能

Model	Accuracy	Precision	Recall	F1-score
ResNet34	0.4778	0.5392	0.4778	0.4714
AdvanceTwoLayerCNN(Ours)	0.4311	0.4580	0.4311	0.4261

在上述表格中，我們清晰地展示了兩個模型在計算成本和性能評估上的比較。從表中可以看出，ResNet34 在性能評估的各個指標上稍稍優於我們的模型，但我們的模型在參數數量上明顯較少，這是其重要優勢之一。

我們的實驗是在 ImageNet-mini 數據集的測試集上進行的。由於算力不足，我們僅訓練了 10 個 epoch，且未使用預訓練模型來訓練 ResNet34。因此，ResNet34 的結果並未達到其最佳性能，但仍能夠為我們的模型提供一個有意義的比較基準。

總結來說，這些實驗結果展示了我們設計的兩層 CNN 網絡在有限層數和資源條件下的有效性。雖然在某些性能指標上稍微落後於 ResNet34，但其在計算成本和參數數量方面具有顯著優勢，表明在特定應用場景中具有很大的潛力。

五、消融實驗

在消融實驗中，我們進行了四種不同配置的模型訓練，觀察其在驗證集上的準確度表現。這四種配置分別是：只有兩層卷積層和一層輸出層的基本模型、僅加入 RRDB 的模型、僅加入自注意力機制的模型，以及我們提出的模型，即兩者並聯加入的情況。礙於算力不足，所有實驗皆在 10 個 epochs 內完成，其他參數設置皆相同。

首先，我們觀察到基本模型只包含兩層卷積層和一層輸出層，在驗證集上的準確度為 0.2244。這個結果表明，即使基本模型僅包含了簡單的卷積層結構，其在一定程度上仍然能夠對圖像進行分類。

表 10 模型架構對於驗證集上準確性的實驗結果

方法	驗證集準確度
使用兩層卷積層和一層輸出層	0.2244
僅加入 RRDB	0.1889
僅加入自注意力機制	0.0956
兩者並聯加入(Ours)	0.2800

接著，我們單獨加入了 RRDB 的模型，在驗證集上的準確度為 0.1889。這顯示出僅使用 RRDB 並不足以提高模型的性能，其準確度反而下降了。這可能是因為在僅進行 10 個 epochs 的訓練情況下，模型尚未能夠充分學習和利用 RRDB 的特性，導致性能下降。另一方面，我們單獨加入了自注意力機制的模型，在驗證集上的準確度為 0.0956。這顯示單獨使用自注意力機制對於提高模型性能並不夠有效，其準確度甚至更低於基本模型。同樣地，這可能是由於訓練的 epoch 數不足，使得模型無法充分利用自注意力機制的優勢。

最後，我們進行了兩者並聯加入的模型訓練，即我們提出的模型。這個模型在驗證集上的準確度為 0.2800，表現最佳。與單獨加入 RRDB 或自注意力機制相比，兩者並聯加入的模型表現更好，其準確度明顯提高。這說明了兩個組件的結合可以相互補充，提高模型的性能。雖然由於計算能力不足，我們僅在有限的 epoch 數下進行了實驗，但結果仍然表明了這種組合的有效性。

總的來說，消融實驗的結果顯示，我們提出的模型在兩個組件並聯加入的情況下達到了最佳性能。雖然單獨使用 RRDB 或自注意力機制無法顯著提高模型性能，但兩者的結合能夠相互補充，使模型性能得到有效提升。在未來的工作中，我們可以進一步探索如何優化模型結構和訓練策略，以更好地利用這些組件的特性，提高模型的性能。

六、結論與展望

在這項研究中，我們成功設計了一個僅含有四層的深度學習模型，在 ImageNet-mini 數據集上實現了 ResNet34 約 90%的性能。值得強調的是，我們僅僅用了這四層的架構，就能夠達到 ResNet34 這樣擁有 34 層的深度神經網絡所具備的性能水平的 90%。通過結合 RRDB 和自注意力機制，實現了在有限層數下的高效圖像分類。雖然我們的模型在某些性能指標上稍微落後於 ResNet34，但在

參數數量方面卻具有明顯的優勢。這表明，我們的設計原則和實現細節在提高模型效率和性能方面具有一定的有效性。

未來，我們可以通過多種方式來進一步優化我們的模型。首先，由於本次實驗受限於計算資源，僅在有限的 epoch 內進行了訓練。如果我們能夠擴大訓練的 epoch 數，可能能夠使模型更好地擬合數據，從而提高性能表現。此外，我們可以進一步優化模型結構，探索更有效的特徵提取方法和注意力機制，以進一步提高準確性和性能。另外，我們可以考慮使用更大的數據集來訓練模型，這將有助於提高模型的泛化能力並改善其在實際應用中的性能。

總的來說，儘管本次實驗受限於算力不足，但我們的研究為深度學習模型在圖像分類領域的應用提供了有價值的參考。未來，我們將繼續努力優化模型，並探索更多新的技術和方法，以提高模型的性能和應用範圍。

伍、 程式碼實現

程式碼實現於 https://github.com/JenLungHsu/DL_assigment2 中，Firstpart 為第一題，動態卷積的實現在 dynamic_conv.py 中，模型實現在 vgg.py 中，訓練時分別訓練 vgg.py 中的 dy_vgg()和 raw_vgg()。測試時也是使用這兩個模型，參數 diff_channel 中可以調整要輸入的通道數為 RGB,RG,GB,R,G,B 中任一值，以達成此次任務。Secondpart 為第二題，自定義模型為 mymodel.py 中的 AdvancedTwoLayerCNN()，訓練結果與 ResNet34 做比較，並於測試集中判斷最終表現結果。

陸、 參考文獻

[1] Wu et al. Dynamic Convolution: Attention over Convolution Kernels. CVPR 2020.

[2] Wang et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. ECCV 2018