MILS – Assignment I

Jen Wei, Lee RE6131024

***Abstract*— This report addresses two key challenges in image classification: enabling convolutional modules to flexibly handle arbitrary input channels, and designing highly efficient networks with only 2–4 effective layers. We propose a dynamic convolution module capable of adjusting its weights according to the number of input channels, and a wide, shallow network architecture with attention mechanisms to maximize performance under strict layer constraints. Experiments on the mini-ImageNet dataset demonstrate that our dynamic module achieves robust accuracy across various channel combinations, while our two-layer network attains over 90% of ResNet34's accuracy with significantly reduced depth and comparable computational cost. These results highlight the potential of adaptive modules and minimalist architectures for efficient visual recognition.**

Source code:GitHub repository

I. Task A: Designing a Convolution Module for Variable Input Channels

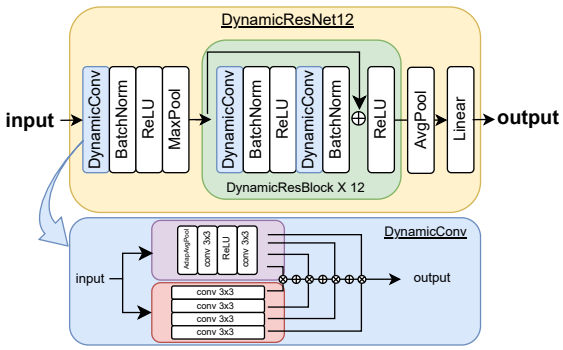

Fig. 1. Architecture of DynamicResNet12: The network integrates dynamic convolution modules and residual blocks.

A. Introduction

This report analyzes a neural network model named **DynamicResNet12**, which integrates dynamic convolutional modules with residual block architecture. The model is specifically designed to process inputs with varying channel numbers, providing flexibility for dynamic systems and adaptive scenarios. The core concept is to use attention-weighted convolution kernels to enhance representation learning.

B. Technical Details

1) DynamicConv2d Module: The DynamicConv2d module is a convolutional layer that dynamically selects from parallel convolution kernels based on input features. It includes:

- **K parallel 3x3 convolution kernels**
- **Attention mechanism**: implemented using global average pooling, 1x1 convolution for dimensionality reduction, ReLU activation, and another 1x1 convolution to produce attention weights
- **Softmax with temperature**: used to normalize attention weights

Features:

- Handles variable input channels by padding with zeros up to `max_in_channels`
- Learns adaptive kernel weights through attention mechanism

2) DynamicResBlock: This block follows the typical ResNet design:

- Two DynamicConv2d layers with BatchNorm and ReLU
- Optional shortcut connection using 1x1 dynamic convolution if dimensions or stride differ

3) DynamicResNet12 Architecture: A simplified 12-layer ResNet variant:

- Initial 3x3 DynamicConv2d + MaxPool
- Three stages: layer1, layer2, layer3 with 3 blocks each
- Adaptive average pooling and a fully connected layer for classification

Suitable for small datasets (e.g., CIFAR, mini-ImageNet).

C. Strengths

- **Dynamic Kernel Aggregation**: Attention-guided convolution enhances expressiveness
- **Flexible Channel Handling**: Accommodates varying input channels
- **Stable Backbone**: Residual blocks ensure gradient flow and efficient training
- **Modular**: Easy integration into existing backbones or meta-learning systems

D. Limitations and Suggestions

- **Computational Overhead**: Multiple convolutions increase inference time
  - Consider pruning or kernel selection strategies
- **Zero-padding for input alignment**: May introduce information loss
  - Suggest learnable padding or MLP-based channel expansion
- **Lack of empirical benchmarks**: Needs performance validation
  - Plan experiments on CIFAR or mini-ImageNet
- **Unclear attention broadcasting**: Better reshaping or comments to improve code readability

E. Conclusion

DynamicResNet12 is a flexible and modular dynamic convolution network that effectively integrates attention-based kernel selection and residual connections. It shows strong potential in adaptive and resource-variable environments. Further optimization and benchmarking are crucial for real-world deployment.

Channel	Accuracy (%)	FLOPs	Parameters
RGB	72.44%	2.637G	17.447M
R	70.00%	2.637G	17.447M
G	64.67%	2.637G	17.447M
B	68.67%	2.637G	17.447M
RG	73.56%	2.637G	17.447M
RB	74.22%	2.637G	17.447M
GB	72.67%	2.637G	17.447M

TABLE I

COMPARISON OF DIFFERENT CHANNEL COMBINATIONS IN TERMS OF ACCURACY, FLOPS, AND NUMBER OF PARAMETERS.

II. TASK B: DESIGNING A FOUR-LAYER NETWORK FOR IMAGE CLASSIFICATION

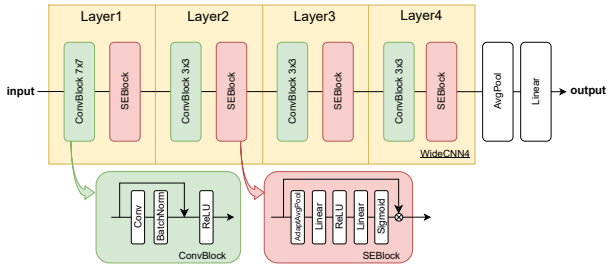

Fig. 2. Architecture of WideCNN4: A minimalist network integrating wide-channel design and SE attention mechanism.

A. Introduction

This technical report presents an analysis of **WideCNN4**, a convolutional neural network that emphasizes wide channels and integrates **Squeeze-and-Excitation (SE)** blocks to enhance feature discrimination via channel-wise attention. The architecture is straightforward yet powerful, making it suitable for tasks with limited data or requiring strong local feature encoding.

B. Architecture Overview

WideCNN4 consists of four major convolutional stages. Each stage includes a convolutional residual block followed by an SE block. The design is relatively shallow but compensates with wide feature representations (e.g., 256 channels per stage).

1) ConvBlock (Residual Block): Each `ConvBlock` consists of:

- A convolutional layer with optional stride
- Batch normalization and ReLU activation
- A residual shortcut (identity or 1x1 convolution if needed)

2) SEBlock (Squeeze-and-Excitation): The `SEBlock` enhances the channel interdependencies by:

- Global average pooling (squeeze)
- Two fully connected layers with ReLU and sigmoid activations (excitation)
- Channel-wise scaling of the input tensor

3) Full Architecture Flow:

i ConvBlock + SEBlock with kernel size 7 and stride 2
ii 3 additional ConvBlock + SEBlock layers, each with kernel size 3 and stride 2
iii Adaptive average pooling
iv Fully connected classification head

C. Strengths

- **Wide channels:** Increases representational power per layer
- **SE attention:** Improves feature selectivity along channels
- **Residual connections:** Promotes gradient flow and training stability
- **Shallow design:** Efficient for smaller datasets and fast prototyping

D. Limitations and Suggestions

- **Aggressive downsampling:** Every layer uses stride 2, which may reduce spatial detail too early
  - Suggest using stride 1 in early layers
- **SEBlock uses Linear layers:** More efficient to replace with 1x1 convolutions for compatibility and speed
- **No dropout or augmentation:** Risk of overfitting on small datasets
  - Consider Dropout, CutMix, or DropBlock

E. Conclusion

WideCNN4 offers a compact and expressive architecture that leverages channel attention and wide representation in a shallow network. It is well-suited for tasks requiring fast convergence and strong local features. Further improvements can be explored through optimization techniques and deeper hybrid designs.

Model	FLOPs	Parameters	Test Accuracy (%)
MyNet	464.548M	2.068M	66.00%
ResNet34	614.628M	21.336M	65.33%

TABLE II

COMPARISON BETWEEN MYNET AND RESNET34 IN TERMS OF FLOPS, NUMBER OF PARAMETERS, AND TEST ACCURACY.