Contents

[I. Presentation of data 2](#_Toc165110163)

[II. Presentation of architectures and historical advancement: 2](#_Toc165110165)

[1. Google Net: 2](#_Toc165110166)

[2. MobileNet V3: 2](#_Toc165110167)

[3. Nasnet: 3](#_Toc165110168)

[III. Transfer learning process walk-through 3](#_Toc165110171)

[IV. Models’ performance and limitations: 4](#_Toc165110183)

[V. Fine-tuning and model training approach: 5](#_Toc165110220)

[VI. Analysis of fine-tuned models’ performance and limitations 6](#_Toc165110236)

[VII. Issues and recommendation 7](#_Toc165110237)

[**APPENDIX** 8](#_Toc165110238)

[1. GoogLeNet architecture: 8](#_Toc165110239)

[2. MobilenetV3 architecture: 20](#_Toc165110240)

[3. NASNetLarge architecture: 32](#_Toc165110241)

[4. GoogLeNet model for training summary: 106](#_Toc165110242)

[5. MobileNetV3 model for training summary: 114](#_Toc165110243)

[6. Nasnet model for training summary: 120](#_Toc165110244)

[7. Fine-tuning Process Details for MobileNetV3 171](#_Toc165110245)

[8. MobileNetV3 Fine tuned model for training summary: 177](#_Toc165110420)

[**REFERENCES** 185](#_Toc165110421)

1. **Presentation of data:**

Food101 is a dataset commonly used for image classification tasks in the field of computer vision. It consists of 101,000 images of food items categorized into 101 classes, each representing a different type of food (such as pizza, sushi, hamburger, etc.). The dataset is widely used for training and testing machine learning models to recognize and classify different types of food items from images.

The data is divided into two sets, including a training set and a test set based on the description of the JSON file. The test set is further split into test and validation sets for evaluation.

1. **Presentation of architectures and historical advancement**:
2. Google Net:

Google Net was published by Google in 2014. It introduced a new concept of inception modules, which help reduce the number of parameters. This module allows the network to capture.

GoogLeNet uses 1x1, 7x7 filters and a 3x3 max pooling. By employing 1x1 convolutions within the inception modules, GoogLeNet reduces the number of parameters while preserving model expressiveness. This reduction helps mitigate overfitting and computational complexity.

The idea behind GoogLeNet is that a hierarchical structure can allow the network to learn complex representations of the input data, thereby leading to better performance in image recognition task. By stacking multiple layers and widening the network, GoogLeNet achieved promising performance on the image classification task without consuming much computational power *(Szegedy et al., 2015)*

Traditional architectures like VGG have many parameters, which make them computationally expensive to train. They also struggled to capture features at multiple scales and levels. By employing filters of various sizes within the same layer, GoogLeNet’s inception modules capture features more effectively, thereby improving its ability to learn complex presentation of data.

1. MobileNet V3:

MobileNet are a family of Neutral Network architecture specifically designed for mobile devices. It is suitable for deployment on devices with limited hardware capabilities.

The first version is the MobileNetV1, which was released in 2017. The model introduced depth wise separable convolutions to reduce number of parameters. Depth wise convolution applies a separate filter to each input channel, resulting in a set of intermediate feature maps. As a result, it facilitates the design of deeper and wider networks within the computational constraints of mobile devices *(Suharjito et al., 2021).*

MobileNetV2 introduces a new building block called "Inverted Residuals with Linear Bottlenecks." It consists of a lightweight bottleneck layer (1x1 pointwise convolution) followed by a depthwise separable convolution and another 1x1 pointwise convolution. Moreover, MobileNetV2 replaces the ReLU activation function in the bottleneck layers with linear activations. Despite the architectural changes, MobileNetV2 maintains compatibility with MobileNetV1 by providing similar functionality and structure.

In 2019, Google researchers introduce MobileNetV3, which contains new architectural elements such as the "MobileNetV3 large" and "MobileNetV3 small" variants *(Babu & B. Sai Chandana, 2023).* They are tailored for different resource constraints and performance requirements. MobileNetV3 introduced several novel architectural changes, including a modified inverted residual structure, squeeze-and-excitation (SE) blocks, and hard swish activation functions. This non-linear activation function is more efficient and less prone to vanishing gradients compared to ReLU. Moreover, novel activation functions such as Hardswish and Hardsigmoid, which are designed to be computationally efficient while still providing non-linearity, help improve the model performance without increasing computational cost. Squeeze-and-Excitation (SE) module enables the network to adaptively recalibrate channel-wise feature responses, enhancing feature representation and improving model performance.

1. Nasnet:

Nasnet architectures represents a sophisticated structure comprising convolutional layers, batch normalization, and branching separable convolutions. NASNet comprises both NASNet-Large and NASNet-Mobile (or Small) variants*.* These variants represent different scales of the architecture designed using neural architecture search (NAS) techniques. NASNET-Large was designed for tasks requiring high accuracy and computational resources, such as image classification. On the other hand, NASNet-Small is suitable for mobile and edge devices with limited computational resources. *(Zoph et al., 2018).*

These architectural elements endow NASNet-Large with the versatility required for effective transfer learning across diverse datasets and tasks. NASNet-Large represents a departure from traditional manual design methodologies. Instead, it leverages neural architecture search (NAS), an automated process that efficiently explores a vast space of possible architectures to identify high-performing models. Thefore, NASNet-Large overcomes the limitations of older models, which relied on human intuition and trial-and-error experimentation to design effective architectures.

1. **Transfer learning process walk-through:**

Transfer learning is the process where a model trained on one task is repurposed on a second related task. It leverages the knowledge gained from the source task to improve learning on the target task, especially when the latter has limited training data.

In the image classification, transfer learning involves using a pre-trained convolutional neural network (CNN) model, which has been trained on a large dataset like ImageNet and fine-tuning it on a smaller dataset like Food101.

The process of transfer learning includes the following steps. First, the pre-trained model architecture is loaded without its classification head. This allows for the reusability of the convolutional base, which has already learned generic features from the source dataset. Second, a new classification head is added to the pre-trained model. In this assignment, we will replace the model head by a Global average layer and three fully connected layers (the thirst one will be the final layer used for making predictions). However, the three models GoogLeNet, MobileNet V3 and Nasnet already have AdaptiveAvgPool2d layer, which effectively performs global average pooling, so it is the same function as a global average layer. As a result, three fully connected layers directly replace the model head. All above models will use cross entropy loss criterion. In Pytorch, torch.nn.CrossEntropyLoss combines the SoftMax activation and the cross-entropy calculation into a single class. So, SoftMax layer is omitted in final layers of all above neutral network architectures. In MobileNet model, a dropout layer is added to reduce the overfitting.

The three added fully connect layers for all three above model is as follow:

***nn.Linear(num\_features, 256), # First fully connected layer:***

***nn.ReLU(),***

***nn.Linear(256, 512), # Second fully connected layer:***

***nn.ReLU(),***

***nn.Linear(512, num\_classes), # Final fully connected layer for classification***

Relu activation function is used because it introduces non-linearity to the network, which is crucial for enabling the model to learn complex relationships in the data. It is also computationally efficient compared to other activation functions like sigmoid or tanh because it involves simple thresholding operations. Moreover, it helps alleviate the vanishing gradient problem, which can occur during backpropagation. It is because ReLU's derivative is either 0 or 1, which prevents the gradient from diminishing too quickly as it propagates through the network, allowing for more stable and efficient training.

The training process are also benefitted from the implementation of a learning rate scheduler. Specifically, the StepLR scheduler was employed to dynamically adjust the learning rate during training. The StepLR scheduler reduces the learning rate by a factor of gamma after a certain number of epochs. This adaptive learning rate scheduling strategy helps to fine-tune the model's optimization process by gradually reducing the learning rate over time. As a result, it promotes more stable convergence and prevents the model from overshooting optimal parameter values during training.

1. **Models’ performance and limitations**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | GoogLeNet | MobilenetV3 | NasNet |
| Accuracy Curve |  |  |  |
| Loss Curve |  |  |  |
| Accuracy | Training: 0.3202  Validation: 0.3415  Testing: 0.3313 | Training: 0.3652  Validation: 0.3611  Testing: 0.3724 | Training: 0.3359  Validation: 0.3015  Testing: 0.3213 |
| Loss | Training: 2.6621  Validation: 2.7518  Testing: 3.0677 | Training: 4.3282  Validation: 4.3360  Testing: 4.3303 | Training: 3.4455  Validation: 3.5589  Testing: 3.5101 |

The GoogLeNet model shows a gradual increase during training, which highlights that the model is learning and improving its performance over epochs. The loss curve shows a decreasing trend during training, indicating that the model is effectively minimizing its loss over epochs. However, the accuracy is low, and the loss is high.

Similarly, the loss curve for MobileNetV3 exhibits a decreasing trend during training. The accuracy and loss values for MobileNetV3 show slightly improved performance compared to GoogLeNet, with slightly higher accuracy and lower loss values.

Nasnet model follows a similar pattern, with accuracy values being slightly lower compared to MobileNetV3. The gap between training and testing accuracy is also larger, which may address overfitting problems.

Overall MobileNetV3, which shows a more consistent performance on both training and testing validation set, with a low difference between training and testing accuracy (0.3652 and 0.3611 respectively), proves to be the best performer among the three models. As a result, it will be chosen to be fine-tuned in the next part.

There are many possible reasons why MobileNetV3 performs better than other models. First, the depth wise separable convolutions of the model are computationally efficient compared to traditional convolutions. Second, inverted residuals with linear bottlenecks facilitate better gradient flow and learning dynamics, which help MobileNetV3 effectively capture complex patterns in the data while mitigating issues like vanishing gradients.

1. **Fine-tuning and model training approach**:

In this part, fine-tuning was performed on the selected best-performing model (MobileNetV3). The fine-tuning process involved freezing a portion of the model's layers while keeping the rest trainable.

Specifically, the first 60 layers of the MobileNetV3 model were frozen to retain the knowledge learned from the ImageNet dataset, while the remaining layers were left trainable to adapt to the Food101 dataset. By freezing a portion of the layers, the fine-tuning process allows the model to retain its ability to recognize generic features learned during pre-training, while simultaneously enabling it to learn task-specific features from the Food101 dataset.

1. **Analysis of fine-tuned models’ performance and limitations:**

|  |  |
| --- | --- |
|  | Fine tuned MobilenetV3 model |
| Accuracy Curve |  |
| Loss Curve |  |
| Accuracy | Training: 0.6862  Validation: 0.7075  Testing: 0.6870 |
| Loss | Training: 1.22  Validation: 1.46  Testing: 2.02 |

There was a notable improvement in both accuracy and loss metrics across all sets. The substantially higher accuracy scores post-fine-tuning demonstrate the model's enhanced ability to correctly classify food items, indicating improved generalization to the nuances of the Food101 dataset.

Furthermore, the considerable reduction in loss values after fine-tuning indicates that the model has become more adept at minimizing prediction errors, resulting in tighter alignment between predicted and actual values during training.

The **Conv2dNormActivation** and **SqueezeExcitation** layers play important roles in the MobileNetV3 architecture. Unfreezing these layers has a notable impact on the model's performance. Batch normalization and activation functions of **Conv2dNormActivation** help in extracting meaningful features from the input data while ensuring that the activations are normalized. It helps reduce the risk of vanishing gradient descents during training, thereby improving model’s performance. On the other hand, the **SqueezeExcitation (SE)** blocks capture channel-wise dependencies in the feature maps. By employing global average pooling followed by fully connected layers and activation functions, the SE blocks learn to capture the importance of different feature channels.

The main limitation is that unfreezing additional layers increases the model's capacity to memorize the training data, which can lead to overfitting. In this experiment, although the performance is improved, the gap between validation and training is widened compared to the previous training model. However, the gap is not too large so there is more room for further development.

1. **Issues and recommendation:**

In conclusion, the MobileNet V3 model has proved to be the best performer. The fine-tuned model of MobileNetV3 significantly improved the models’ performance. However, the widening gap between validation and training performance typically indicates an increase in overfitting.

The possible solution for this issue is to apply regularization techniques and experiment with different model architectures and hyperparameters to find the best balance between model complexity and generalization.

# **APPENDIX**

1. **GoogLeNet architecture**:

GoogLeNet(

(conv1): BasicConv2d(

(conv): Conv2d(3, 64, kernel\_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(maxpool1): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=True)

(conv2): BasicConv2d(

(conv): Conv2d(64, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv3): BasicConv2d(

(conv): Conv2d(64, 192, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(maxpool2): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=True)

(inception3a): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(192, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(192, 96, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(96, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(96, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(192, 16, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(16, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(16, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(192, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(inception3b): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(128, 192, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(32, 96, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(96, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(256, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(maxpool3): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=True)

(inception4a): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(480, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(480, 96, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(96, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(96, 208, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(208, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(480, 16, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(16, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(16, 48, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(480, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(inception4b): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(512, 160, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(512, 112, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(112, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(112, 224, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(224, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(512, 24, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(24, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(512, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(inception4c): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(512, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(512, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(512, 24, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(24, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(512, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(inception4d): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(512, 112, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(112, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(512, 144, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(144, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(144, 288, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(288, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(512, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(512, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(inception4e): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(528, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(528, 160, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(160, 320, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(320, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(528, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(32, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(528, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(maxpool4): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=True)

(inception5a): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(832, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(832, 160, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(160, 320, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(320, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(832, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(32, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(832, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(inception5b): Inception(

(branch1): BasicConv2d(

(conv): Conv2d(832, 384, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(384, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(832, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(192, 384, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(384, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch3): Sequential(

(0): BasicConv2d(

(conv): Conv2d(832, 48, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(1): BasicConv2d(

(conv): Conv2d(48, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(branch4): Sequential(

(0): MaxPool2d(kernel\_size=3, stride=1, padding=1, dilation=1, ceil\_mode=True)

(1): BasicConv2d(

(conv): Conv2d(832, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

)

(aux1): None

(aux2): None

(avgpool): AdaptiveAvgPool2d(output\_size=(1, 1))

(dropout): Dropout(p=0.2, inplace=False)

(fc): Linear(in\_features=1024, out\_features=1000, bias=True)

)

1. **MobilenetV3 architecture:**

MobileNetV3(

(features): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(3, 16, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(16, 16, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=16, bias=False)

(1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(1): Conv2dNormActivation(

(0): Conv2d(16, 16, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(2): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(16, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(1): Conv2dNormActivation(

(0): Conv2d(64, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=64, bias=False)

(1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(2): Conv2dNormActivation(

(0): Conv2d(64, 24, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(3): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(24, 72, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(1): Conv2dNormActivation(

(0): Conv2d(72, 72, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=72, bias=False)

(1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(2): Conv2dNormActivation(

(0): Conv2d(72, 24, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(4): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(24, 72, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(1): Conv2dNormActivation(

(0): Conv2d(72, 72, kernel\_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=72, bias=False)

(1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(72, 24, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(24, 72, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(72, 40, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(5): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(40, 120, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(1): Conv2dNormActivation(

(0): Conv2d(120, 120, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=120, bias=False)

(1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(120, 32, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(32, 120, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(120, 40, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(6): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(40, 120, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(1): Conv2dNormActivation(

(0): Conv2d(120, 120, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=120, bias=False)

(1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(120, 32, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(32, 120, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(120, 40, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(7): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(40, 240, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(240, 240, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=240, bias=False)

(1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): Conv2dNormActivation(

(0): Conv2d(240, 80, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(8): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(80, 200, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(200, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(200, 200, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=200, bias=False)

(1): BatchNorm2d(200, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): Conv2dNormActivation(

(0): Conv2d(200, 80, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(9): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(80, 184, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(184, 184, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=184, bias=False)

(1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): Conv2dNormActivation(

(0): Conv2d(184, 80, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(10): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(80, 184, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(184, 184, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=184, bias=False)

(1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): Conv2dNormActivation(

(0): Conv2d(184, 80, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(11): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(80, 480, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(480, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(480, 480, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)

(1): BatchNorm2d(480, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(480, 120, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(120, 480, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(480, 112, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(112, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(12): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(112, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(672, 168, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(168, 672, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(672, 112, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(112, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(13): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(112, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(672, 672, kernel\_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=672, bias=False)

(1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(672, 168, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(168, 672, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(672, 160, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(14): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(160, 960, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(960, 960, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=960, bias=False)

(1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(960, 240, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(240, 960, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(960, 160, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(15): InvertedResidual(

(block): Sequential(

(0): Conv2dNormActivation(

(0): Conv2d(160, 960, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(1): Conv2dNormActivation(

(0): Conv2d(960, 960, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=960, bias=False)

(1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

(2): SqueezeExcitation(

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(fc1): Conv2d(960, 240, kernel\_size=(1, 1), stride=(1, 1))

(fc2): Conv2d(240, 960, kernel\_size=(1, 1), stride=(1, 1))

(activation): ReLU()

(scale\_activation): Hardsigmoid()

)

(3): Conv2dNormActivation(

(0): Conv2d(960, 160, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

)

)

)

(16): Conv2dNormActivation(

(0): Conv2d(160, 960, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True, track\_running\_stats=True)

(2): Hardswish()

)

)

(avgpool): AdaptiveAvgPool2d(output\_size=1)

(classifier): Sequential(

(0): Linear(in\_features=960, out\_features=1280, bias=True)

(1): Hardswish()

(2): Dropout(p=0.2, inplace=True)

(3): Linear(in\_features=1280, out\_features=1000, bias=True)

)

)

1. **NASNetLarge architecture:**

NASNetALarge(

(conv0): ConvNormAct(

(conv): Conv2d(3, 96, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNormAct2d(

96, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True

(drop): Identity()

(act): Identity()

)

)

(cell\_stem\_0): CellStem0(

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(96, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(42, 42, kernel\_size=(5, 5), stride=(2, 2), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(42, 42, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(96, 96, kernel\_size=(7, 7), stride=(2, 2), groups=96, bias=False)

(pointwise\_conv2d): Conv2d(96, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(42, 42, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(96, 96, kernel\_size=(7, 7), stride=(2, 2), groups=96, bias=False)

(pointwise\_conv2d): Conv2d(96, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(42, 42, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0))

(comb\_iter\_2\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(96, 96, kernel\_size=(5, 5), stride=(2, 2), groups=96, bias=False)

(pointwise\_conv2d): Conv2d(96, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(42, 42, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(42, 42, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(42, 42, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=42, bias=False)

(pointwise\_conv2d): Conv2d(42, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(42, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_4\_right): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

)

(cell\_stem\_1): CellStem1(

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(168, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(act): ReLU()

(path\_1): Sequential(

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(96, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(path\_2): Sequential(

(pad): ZeroPad2d((-1, 1, -1, 1))

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(96, 42, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(final\_path\_bn): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(84, 84, kernel\_size=(5, 5), stride=(2, 2), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(84, 84, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(84, 84, kernel\_size=(7, 7), stride=(2, 2), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(84, 84, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(84, 84, kernel\_size=(7, 7), stride=(2, 2), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(84, 84, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0))

(comb\_iter\_2\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(84, 84, kernel\_size=(5, 5), stride=(2, 2), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(84, 84, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(84, 84, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(84, 84, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=84, bias=False)

(pointwise\_conv2d): Conv2d(84, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(84, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_4\_right): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

)

(cell\_0): FirstCell(

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(336, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(act): ReLU()

(path\_1): Sequential(

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(168, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(path\_2): Sequential(

(pad): ZeroPad2d((-1, 1, -1, 1))

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(168, 84, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(final\_path\_bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_1): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(336, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_2): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_3): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_4): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_5): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(168, 168, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=168, bias=False)

(pointwise\_conv2d): Conv2d(168, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(168, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(reduction\_cell\_0): ReductionCell0(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1008, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(336, 336, kernel\_size=(5, 5), stride=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(336, 336, kernel\_size=(7, 7), stride=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(336, 336, kernel\_size=(7, 7), stride=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0))

(comb\_iter\_2\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(336, 336, kernel\_size=(5, 5), stride=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_4\_right): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

)

(cell\_6): FirstCell(

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1344, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(act): ReLU()

(path\_1): Sequential(

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(path\_2): Sequential(

(pad): ZeroPad2d((-1, 1, -1, 1))

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(1008, 168, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(final\_path\_bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_7): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(1344, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_8): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_9): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_10): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_11): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(336, 336, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)

(pointwise\_conv2d): Conv2d(336, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(336, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(reduction\_cell\_1): ReductionCell1(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2016, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(672, 672, kernel\_size=(5, 5), stride=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(672, 672, kernel\_size=(7, 7), stride=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(672, 672, kernel\_size=(7, 7), stride=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(7, 7), stride=(1, 1), padding=(3, 3), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0))

(comb\_iter\_2\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2dSame(672, 672, kernel\_size=(5, 5), stride=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_4\_right): MaxPool2dSame(kernel\_size=(3, 3), stride=(2, 2), padding=(0, 0), dilation=(1, 1), ceil\_mode=False)

)

(cell\_12): FirstCell(

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2688, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(act): ReLU()

(path\_1): Sequential(

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(path\_2): Sequential(

(pad): ZeroPad2d((-1, 1, -1, 1))

(avgpool): AvgPool2d(kernel\_size=1, stride=2, padding=0)

(conv): Conv2d(2016, 336, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(final\_path\_bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_13): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(2688, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_14): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_15): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_16): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(cell\_17): NormalCell(

(conv\_prev\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(conv\_1x1): ActConvBn(

(act): ReLU()

(conv): Conv2d(4032, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_0\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_1\_right): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

(comb\_iter\_2\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_left): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_3\_right): AvgPool2d(kernel\_size=3, stride=1, padding=1)

(comb\_iter\_4\_left): BranchSeparables(

(act\_1): ReLU()

(separable\_1): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_1): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(act\_2): ReLU(inplace=True)

(separable\_2): SeparableConv2d(

(depthwise\_conv2d): Conv2d(672, 672, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672, bias=False)

(pointwise\_conv2d): Conv2d(672, 672, kernel\_size=(1, 1), stride=(1, 1), bias=False)

)

(bn\_sep\_2): BatchNorm2d(672, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

)

)

(act): ReLU(inplace=True)

(global\_pool): SelectAdaptivePool2d(pool\_type=avg, flatten=Flatten(start\_dim=1, end\_dim=-1))

(head\_drop): Dropout(p=0.0, inplace=False)

(last\_linear): Linear(in\_features=4032, out\_features=1000, bias=True)

)

1. GoogLeNet model for training summary:

----------------------------------------------------------------

Layer (type) Output Shape Param #

================================================================

Conv2d-1 [-1, 64, 112, 112] 9,408

BatchNorm2d-2 [-1, 64, 112, 112] 128

BasicConv2d-3 [-1, 64, 112, 112] 0

MaxPool2d-4 [-1, 64, 56, 56] 0

Conv2d-5 [-1, 64, 56, 56] 4,096

BatchNorm2d-6 [-1, 64, 56, 56] 128

BasicConv2d-7 [-1, 64, 56, 56] 0

Conv2d-8 [-1, 192, 56, 56] 110,592

BatchNorm2d-9 [-1, 192, 56, 56] 384

BasicConv2d-10 [-1, 192, 56, 56] 0

MaxPool2d-11 [-1, 192, 28, 28] 0

Conv2d-12 [-1, 64, 28, 28] 12,288

BatchNorm2d-13 [-1, 64, 28, 28] 128

BasicConv2d-14 [-1, 64, 28, 28] 0

Conv2d-15 [-1, 96, 28, 28] 18,432

BatchNorm2d-16 [-1, 96, 28, 28] 192

BasicConv2d-17 [-1, 96, 28, 28] 0

Conv2d-18 [-1, 128, 28, 28] 110,592

BatchNorm2d-19 [-1, 128, 28, 28] 256

BasicConv2d-20 [-1, 128, 28, 28] 0

Conv2d-21 [-1, 16, 28, 28] 3,072

BatchNorm2d-22 [-1, 16, 28, 28] 32

BasicConv2d-23 [-1, 16, 28, 28] 0

Conv2d-24 [-1, 32, 28, 28] 4,608

BatchNorm2d-25 [-1, 32, 28, 28] 64

BasicConv2d-26 [-1, 32, 28, 28] 0

MaxPool2d-27 [-1, 192, 28, 28] 0

Conv2d-28 [-1, 32, 28, 28] 6,144

BatchNorm2d-29 [-1, 32, 28, 28] 64

BasicConv2d-30 [-1, 32, 28, 28] 0

Inception-31 [-1, 256, 28, 28] 0

Conv2d-32 [-1, 128, 28, 28] 32,768

BatchNorm2d-33 [-1, 128, 28, 28] 256

BasicConv2d-34 [-1, 128, 28, 28] 0

Conv2d-35 [-1, 128, 28, 28] 32,768

BatchNorm2d-36 [-1, 128, 28, 28] 256

BasicConv2d-37 [-1, 128, 28, 28] 0

Conv2d-38 [-1, 192, 28, 28] 221,184

BatchNorm2d-39 [-1, 192, 28, 28] 384

BasicConv2d-40 [-1, 192, 28, 28] 0

Conv2d-41 [-1, 32, 28, 28] 8,192

BatchNorm2d-42 [-1, 32, 28, 28] 64

BasicConv2d-43 [-1, 32, 28, 28] 0

Conv2d-44 [-1, 96, 28, 28] 27,648

BatchNorm2d-45 [-1, 96, 28, 28] 192

BasicConv2d-46 [-1, 96, 28, 28] 0

MaxPool2d-47 [-1, 256, 28, 28] 0

Conv2d-48 [-1, 64, 28, 28] 16,384

BatchNorm2d-49 [-1, 64, 28, 28] 128

BasicConv2d-50 [-1, 64, 28, 28] 0

Inception-51 [-1, 480, 28, 28] 0

MaxPool2d-52 [-1, 480, 14, 14] 0

Conv2d-53 [-1, 192, 14, 14] 92,160

BatchNorm2d-54 [-1, 192, 14, 14] 384

BasicConv2d-55 [-1, 192, 14, 14] 0

Conv2d-56 [-1, 96, 14, 14] 46,080

BatchNorm2d-57 [-1, 96, 14, 14] 192

BasicConv2d-58 [-1, 96, 14, 14] 0

Conv2d-59 [-1, 208, 14, 14] 179,712

BatchNorm2d-60 [-1, 208, 14, 14] 416

BasicConv2d-61 [-1, 208, 14, 14] 0

Conv2d-62 [-1, 16, 14, 14] 7,680

BatchNorm2d-63 [-1, 16, 14, 14] 32

BasicConv2d-64 [-1, 16, 14, 14] 0

Conv2d-65 [-1, 48, 14, 14] 6,912

BatchNorm2d-66 [-1, 48, 14, 14] 96

BasicConv2d-67 [-1, 48, 14, 14] 0

MaxPool2d-68 [-1, 480, 14, 14] 0

Conv2d-69 [-1, 64, 14, 14] 30,720

BatchNorm2d-70 [-1, 64, 14, 14] 128

BasicConv2d-71 [-1, 64, 14, 14] 0

Inception-72 [-1, 512, 14, 14] 0

Conv2d-73 [-1, 160, 14, 14] 81,920

BatchNorm2d-74 [-1, 160, 14, 14] 320

BasicConv2d-75 [-1, 160, 14, 14] 0

Conv2d-76 [-1, 112, 14, 14] 57,344

BatchNorm2d-77 [-1, 112, 14, 14] 224

BasicConv2d-78 [-1, 112, 14, 14] 0

Conv2d-79 [-1, 224, 14, 14] 225,792

BatchNorm2d-80 [-1, 224, 14, 14] 448

BasicConv2d-81 [-1, 224, 14, 14] 0

Conv2d-82 [-1, 24, 14, 14] 12,288

BatchNorm2d-83 [-1, 24, 14, 14] 48

BasicConv2d-84 [-1, 24, 14, 14] 0

Conv2d-85 [-1, 64, 14, 14] 13,824

BatchNorm2d-86 [-1, 64, 14, 14] 128

BasicConv2d-87 [-1, 64, 14, 14] 0

MaxPool2d-88 [-1, 512, 14, 14] 0

Conv2d-89 [-1, 64, 14, 14] 32,768

BatchNorm2d-90 [-1, 64, 14, 14] 128

BasicConv2d-91 [-1, 64, 14, 14] 0

Inception-92 [-1, 512, 14, 14] 0

Conv2d-93 [-1, 128, 14, 14] 65,536

BatchNorm2d-94 [-1, 128, 14, 14] 256

BasicConv2d-95 [-1, 128, 14, 14] 0

Conv2d-96 [-1, 128, 14, 14] 65,536

BatchNorm2d-97 [-1, 128, 14, 14] 256

BasicConv2d-98 [-1, 128, 14, 14] 0

Conv2d-99 [-1, 256, 14, 14] 294,912

BatchNorm2d-100 [-1, 256, 14, 14] 512

BasicConv2d-101 [-1, 256, 14, 14] 0

Conv2d-102 [-1, 24, 14, 14] 12,288

BatchNorm2d-103 [-1, 24, 14, 14] 48

BasicConv2d-104 [-1, 24, 14, 14] 0

Conv2d-105 [-1, 64, 14, 14] 13,824

BatchNorm2d-106 [-1, 64, 14, 14] 128

BasicConv2d-107 [-1, 64, 14, 14] 0

MaxPool2d-108 [-1, 512, 14, 14] 0

Conv2d-109 [-1, 64, 14, 14] 32,768

BatchNorm2d-110 [-1, 64, 14, 14] 128

BasicConv2d-111 [-1, 64, 14, 14] 0

Inception-112 [-1, 512, 14, 14] 0

Conv2d-113 [-1, 112, 14, 14] 57,344

BatchNorm2d-114 [-1, 112, 14, 14] 224

BasicConv2d-115 [-1, 112, 14, 14] 0

Conv2d-116 [-1, 144, 14, 14] 73,728

BatchNorm2d-117 [-1, 144, 14, 14] 288

BasicConv2d-118 [-1, 144, 14, 14] 0

Conv2d-119 [-1, 288, 14, 14] 373,248

BatchNorm2d-120 [-1, 288, 14, 14] 576

BasicConv2d-121 [-1, 288, 14, 14] 0

Conv2d-122 [-1, 32, 14, 14] 16,384

BatchNorm2d-123 [-1, 32, 14, 14] 64

BasicConv2d-124 [-1, 32, 14, 14] 0

Conv2d-125 [-1, 64, 14, 14] 18,432

BatchNorm2d-126 [-1, 64, 14, 14] 128

BasicConv2d-127 [-1, 64, 14, 14] 0

MaxPool2d-128 [-1, 512, 14, 14] 0

Conv2d-129 [-1, 64, 14, 14] 32,768

BatchNorm2d-130 [-1, 64, 14, 14] 128

BasicConv2d-131 [-1, 64, 14, 14] 0

Inception-132 [-1, 528, 14, 14] 0

Conv2d-133 [-1, 256, 14, 14] 135,168

BatchNorm2d-134 [-1, 256, 14, 14] 512

BasicConv2d-135 [-1, 256, 14, 14] 0

Conv2d-136 [-1, 160, 14, 14] 84,480

BatchNorm2d-137 [-1, 160, 14, 14] 320

BasicConv2d-138 [-1, 160, 14, 14] 0

Conv2d-139 [-1, 320, 14, 14] 460,800

BatchNorm2d-140 [-1, 320, 14, 14] 640

BasicConv2d-141 [-1, 320, 14, 14] 0

Conv2d-142 [-1, 32, 14, 14] 16,896

BatchNorm2d-143 [-1, 32, 14, 14] 64

BasicConv2d-144 [-1, 32, 14, 14] 0

Conv2d-145 [-1, 128, 14, 14] 36,864

BatchNorm2d-146 [-1, 128, 14, 14] 256

BasicConv2d-147 [-1, 128, 14, 14] 0

MaxPool2d-148 [-1, 528, 14, 14] 0

Conv2d-149 [-1, 128, 14, 14] 67,584

BatchNorm2d-150 [-1, 128, 14, 14] 256

BasicConv2d-151 [-1, 128, 14, 14] 0

Inception-152 [-1, 832, 14, 14] 0

MaxPool2d-153 [-1, 832, 7, 7] 0

Conv2d-154 [-1, 256, 7, 7] 212,992

BatchNorm2d-155 [-1, 256, 7, 7] 512

BasicConv2d-156 [-1, 256, 7, 7] 0

Conv2d-157 [-1, 160, 7, 7] 133,120

BatchNorm2d-158 [-1, 160, 7, 7] 320

BasicConv2d-159 [-1, 160, 7, 7] 0

Conv2d-160 [-1, 320, 7, 7] 460,800

BatchNorm2d-161 [-1, 320, 7, 7] 640

BasicConv2d-162 [-1, 320, 7, 7] 0

Conv2d-163 [-1, 32, 7, 7] 26,624

BatchNorm2d-164 [-1, 32, 7, 7] 64

BasicConv2d-165 [-1, 32, 7, 7] 0

Conv2d-166 [-1, 128, 7, 7] 36,864

BatchNorm2d-167 [-1, 128, 7, 7] 256

BasicConv2d-168 [-1, 128, 7, 7] 0

MaxPool2d-169 [-1, 832, 7, 7] 0

Conv2d-170 [-1, 128, 7, 7] 106,496

BatchNorm2d-171 [-1, 128, 7, 7] 256

BasicConv2d-172 [-1, 128, 7, 7] 0

Inception-173 [-1, 832, 7, 7] 0

Conv2d-174 [-1, 384, 7, 7] 319,488

BatchNorm2d-175 [-1, 384, 7, 7] 768

BasicConv2d-176 [-1, 384, 7, 7] 0

Conv2d-177 [-1, 192, 7, 7] 159,744

BatchNorm2d-178 [-1, 192, 7, 7] 384

BasicConv2d-179 [-1, 192, 7, 7] 0

Conv2d-180 [-1, 384, 7, 7] 663,552

BatchNorm2d-181 [-1, 384, 7, 7] 768

BasicConv2d-182 [-1, 384, 7, 7] 0

Conv2d-183 [-1, 48, 7, 7] 39,936

BatchNorm2d-184 [-1, 48, 7, 7] 96

BasicConv2d-185 [-1, 48, 7, 7] 0

Conv2d-186 [-1, 128, 7, 7] 55,296

BatchNorm2d-187 [-1, 128, 7, 7] 256

BasicConv2d-188 [-1, 128, 7, 7] 0

MaxPool2d-189 [-1, 832, 7, 7] 0

Conv2d-190 [-1, 128, 7, 7] 106,496

BatchNorm2d-191 [-1, 128, 7, 7] 256

BasicConv2d-192 [-1, 128, 7, 7] 0

Inception-193 [-1, 1024, 7, 7] 0

AdaptiveAvgPool2d-194 [-1, 1024, 1, 1] 0

Dropout-195 [-1, 1024] 0

Linear-196 [-1, 256] 262,400

ReLU-197 [-1, 256] 0

Linear-198 [-1, 512] 131,584

ReLU-199 [-1, 512] 0

Linear-200 [-1, 101] 51,813

================================================================

Total params: 6,045,701

Trainable params: 445,797

Non-trainable params: 5,599,904

----------------------------------------------------------------

Input size (MB): 0.57

Forward/backward pass size (MB): 94.11

Params size (MB): 23.06

Estimated Total Size (MB): 117.75

----------------------------------------------------------------

1. MobileNetV3 model for training summary:

----------------------------------------------------------------

Layer (type) Output Shape Param #

================================================================

Conv2d-1 [-1, 16, 112, 112] 432

BatchNorm2d-2 [-1, 16, 112, 112] 32

Hardswish-3 [-1, 16, 112, 112] 0

Conv2d-4 [-1, 16, 112, 112] 144

BatchNorm2d-5 [-1, 16, 112, 112] 32

ReLU-6 [-1, 16, 112, 112] 0

Conv2d-7 [-1, 16, 112, 112] 256

BatchNorm2d-8 [-1, 16, 112, 112] 32

InvertedResidual-9 [-1, 16, 112, 112] 0

Conv2d-10 [-1, 64, 112, 112] 1,024

BatchNorm2d-11 [-1, 64, 112, 112] 128

ReLU-12 [-1, 64, 112, 112] 0

Conv2d-13 [-1, 64, 56, 56] 576

BatchNorm2d-14 [-1, 64, 56, 56] 128

ReLU-15 [-1, 64, 56, 56] 0

Conv2d-16 [-1, 24, 56, 56] 1,536

BatchNorm2d-17 [-1, 24, 56, 56] 48

InvertedResidual-18 [-1, 24, 56, 56] 0

Conv2d-19 [-1, 72, 56, 56] 1,728

BatchNorm2d-20 [-1, 72, 56, 56] 144

ReLU-21 [-1, 72, 56, 56] 0

Conv2d-22 [-1, 72, 56, 56] 648

BatchNorm2d-23 [-1, 72, 56, 56] 144

ReLU-24 [-1, 72, 56, 56] 0

Conv2d-25 [-1, 24, 56, 56] 1,728

BatchNorm2d-26 [-1, 24, 56, 56] 48

InvertedResidual-27 [-1, 24, 56, 56] 0

Conv2d-28 [-1, 72, 56, 56] 1,728

BatchNorm2d-29 [-1, 72, 56, 56] 144

ReLU-30 [-1, 72, 56, 56] 0

Conv2d-31 [-1, 72, 28, 28] 1,800

BatchNorm2d-32 [-1, 72, 28, 28] 144

ReLU-33 [-1, 72, 28, 28] 0

AdaptiveAvgPool2d-34 [-1, 72, 1, 1] 0

Conv2d-35 [-1, 24, 1, 1] 1,752

ReLU-36 [-1, 24, 1, 1] 0

Conv2d-37 [-1, 72, 1, 1] 1,800

Hardsigmoid-38 [-1, 72, 1, 1] 0

SqueezeExcitation-39 [-1, 72, 28, 28] 0

Conv2d-40 [-1, 40, 28, 28] 2,880

BatchNorm2d-41 [-1, 40, 28, 28] 80

InvertedResidual-42 [-1, 40, 28, 28] 0

Conv2d-43 [-1, 120, 28, 28] 4,800

BatchNorm2d-44 [-1, 120, 28, 28] 240

ReLU-45 [-1, 120, 28, 28] 0

Conv2d-46 [-1, 120, 28, 28] 3,000

BatchNorm2d-47 [-1, 120, 28, 28] 240

ReLU-48 [-1, 120, 28, 28] 0

AdaptiveAvgPool2d-49 [-1, 120, 1, 1] 0

Conv2d-50 [-1, 32, 1, 1] 3,872

ReLU-51 [-1, 32, 1, 1] 0

Conv2d-52 [-1, 120, 1, 1] 3,960

Hardsigmoid-53 [-1, 120, 1, 1] 0

SqueezeExcitation-54 [-1, 120, 28, 28] 0

Conv2d-55 [-1, 40, 28, 28] 4,800

BatchNorm2d-56 [-1, 40, 28, 28] 80

InvertedResidual-57 [-1, 40, 28, 28] 0

Conv2d-58 [-1, 120, 28, 28] 4,800

BatchNorm2d-59 [-1, 120, 28, 28] 240

ReLU-60 [-1, 120, 28, 28] 0

Conv2d-61 [-1, 120, 28, 28] 3,000

BatchNorm2d-62 [-1, 120, 28, 28] 240

ReLU-63 [-1, 120, 28, 28] 0

AdaptiveAvgPool2d-64 [-1, 120, 1, 1] 0

Conv2d-65 [-1, 32, 1, 1] 3,872

ReLU-66 [-1, 32, 1, 1] 0

Conv2d-67 [-1, 120, 1, 1] 3,960

Hardsigmoid-68 [-1, 120, 1, 1] 0

SqueezeExcitation-69 [-1, 120, 28, 28] 0

Conv2d-70 [-1, 40, 28, 28] 4,800

BatchNorm2d-71 [-1, 40, 28, 28] 80

InvertedResidual-72 [-1, 40, 28, 28] 0

Conv2d-73 [-1, 240, 28, 28] 9,600

BatchNorm2d-74 [-1, 240, 28, 28] 480

Hardswish-75 [-1, 240, 28, 28] 0

Conv2d-76 [-1, 240, 14, 14] 2,160

BatchNorm2d-77 [-1, 240, 14, 14] 480

Hardswish-78 [-1, 240, 14, 14] 0

Conv2d-79 [-1, 80, 14, 14] 19,200

BatchNorm2d-80 [-1, 80, 14, 14] 160

InvertedResidual-81 [-1, 80, 14, 14] 0

Conv2d-82 [-1, 200, 14, 14] 16,000

BatchNorm2d-83 [-1, 200, 14, 14] 400

Hardswish-84 [-1, 200, 14, 14] 0

Conv2d-85 [-1, 200, 14, 14] 1,800

BatchNorm2d-86 [-1, 200, 14, 14] 400

Hardswish-87 [-1, 200, 14, 14] 0

Conv2d-88 [-1, 80, 14, 14] 16,000

BatchNorm2d-89 [-1, 80, 14, 14] 160

InvertedResidual-90 [-1, 80, 14, 14] 0

Conv2d-91 [-1, 184, 14, 14] 14,720

BatchNorm2d-92 [-1, 184, 14, 14] 368

Hardswish-93 [-1, 184, 14, 14] 0

Conv2d-94 [-1, 184, 14, 14] 1,656

BatchNorm2d-95 [-1, 184, 14, 14] 368

Hardswish-96 [-1, 184, 14, 14] 0

Conv2d-97 [-1, 80, 14, 14] 14,720

BatchNorm2d-98 [-1, 80, 14, 14] 160

InvertedResidual-99 [-1, 80, 14, 14] 0

Conv2d-100 [-1, 184, 14, 14] 14,720

BatchNorm2d-101 [-1, 184, 14, 14] 368

Hardswish-102 [-1, 184, 14, 14] 0

Conv2d-103 [-1, 184, 14, 14] 1,656

BatchNorm2d-104 [-1, 184, 14, 14] 368

Hardswish-105 [-1, 184, 14, 14] 0

Conv2d-106 [-1, 80, 14, 14] 14,720

BatchNorm2d-107 [-1, 80, 14, 14] 160

InvertedResidual-108 [-1, 80, 14, 14] 0

Conv2d-109 [-1, 480, 14, 14] 38,400

BatchNorm2d-110 [-1, 480, 14, 14] 960

Hardswish-111 [-1, 480, 14, 14] 0

Conv2d-112 [-1, 480, 14, 14] 4,320

BatchNorm2d-113 [-1, 480, 14, 14] 960

Hardswish-114 [-1, 480, 14, 14] 0

AdaptiveAvgPool2d-115 [-1, 480, 1, 1] 0

Conv2d-116 [-1, 120, 1, 1] 57,720

ReLU-117 [-1, 120, 1, 1] 0

Conv2d-118 [-1, 480, 1, 1] 58,080

Hardsigmoid-119 [-1, 480, 1, 1] 0

SqueezeExcitation-120 [-1, 480, 14, 14] 0

Conv2d-121 [-1, 112, 14, 14] 53,760

BatchNorm2d-122 [-1, 112, 14, 14] 224

InvertedResidual-123 [-1, 112, 14, 14] 0

Conv2d-124 [-1, 672, 14, 14] 75,264

BatchNorm2d-125 [-1, 672, 14, 14] 1,344

Hardswish-126 [-1, 672, 14, 14] 0

Conv2d-127 [-1, 672, 14, 14] 6,048

BatchNorm2d-128 [-1, 672, 14, 14] 1,344

Hardswish-129 [-1, 672, 14, 14] 0

AdaptiveAvgPool2d-130 [-1, 672, 1, 1] 0

Conv2d-131 [-1, 168, 1, 1] 113,064

ReLU-132 [-1, 168, 1, 1] 0

Conv2d-133 [-1, 672, 1, 1] 113,568

Hardsigmoid-134 [-1, 672, 1, 1] 0

SqueezeExcitation-135 [-1, 672, 14, 14] 0

Conv2d-136 [-1, 112, 14, 14] 75,264

BatchNorm2d-137 [-1, 112, 14, 14] 224

InvertedResidual-138 [-1, 112, 14, 14] 0

Conv2d-139 [-1, 672, 14, 14] 75,264

BatchNorm2d-140 [-1, 672, 14, 14] 1,344

Hardswish-141 [-1, 672, 14, 14] 0

Conv2d-142 [-1, 672, 7, 7] 16,800

BatchNorm2d-143 [-1, 672, 7, 7] 1,344

Hardswish-144 [-1, 672, 7, 7] 0

AdaptiveAvgPool2d-145 [-1, 672, 1, 1] 0

Conv2d-146 [-1, 168, 1, 1] 113,064

ReLU-147 [-1, 168, 1, 1] 0

Conv2d-148 [-1, 672, 1, 1] 113,568

Hardsigmoid-149 [-1, 672, 1, 1] 0

SqueezeExcitation-150 [-1, 672, 7, 7] 0

Conv2d-151 [-1, 160, 7, 7] 107,520

BatchNorm2d-152 [-1, 160, 7, 7] 320

InvertedResidual-153 [-1, 160, 7, 7] 0

Conv2d-154 [-1, 960, 7, 7] 153,600

BatchNorm2d-155 [-1, 960, 7, 7] 1,920

Hardswish-156 [-1, 960, 7, 7] 0

Conv2d-157 [-1, 960, 7, 7] 24,000

BatchNorm2d-158 [-1, 960, 7, 7] 1,920

Hardswish-159 [-1, 960, 7, 7] 0

AdaptiveAvgPool2d-160 [-1, 960, 1, 1] 0

Conv2d-161 [-1, 240, 1, 1] 230,640

ReLU-162 [-1, 240, 1, 1] 0

Conv2d-163 [-1, 960, 1, 1] 231,360

Hardsigmoid-164 [-1, 960, 1, 1] 0

SqueezeExcitation-165 [-1, 960, 7, 7] 0

Conv2d-166 [-1, 160, 7, 7] 153,600

BatchNorm2d-167 [-1, 160, 7, 7] 320

InvertedResidual-168 [-1, 160, 7, 7] 0

Conv2d-169 [-1, 960, 7, 7] 153,600

BatchNorm2d-170 [-1, 960, 7, 7] 1,920

Hardswish-171 [-1, 960, 7, 7] 0

Conv2d-172 [-1, 960, 7, 7] 24,000

BatchNorm2d-173 [-1, 960, 7, 7] 1,920

Hardswish-174 [-1, 960, 7, 7] 0

AdaptiveAvgPool2d-175 [-1, 960, 1, 1] 0

Conv2d-176 [-1, 240, 1, 1] 230,640

ReLU-177 [-1, 240, 1, 1] 0

Conv2d-178 [-1, 960, 1, 1] 231,360

Hardsigmoid-179 [-1, 960, 1, 1] 0

SqueezeExcitation-180 [-1, 960, 7, 7] 0

Conv2d-181 [-1, 160, 7, 7] 153,600

BatchNorm2d-182 [-1, 160, 7, 7] 320

InvertedResidual-183 [-1, 160, 7, 7] 0

Conv2d-184 [-1, 960, 7, 7] 153,600

BatchNorm2d-185 [-1, 960, 7, 7] 1,920

Hardswish-186 [-1, 960, 7, 7] 0

AdaptiveAvgPool2d-187 [-1, 960, 1, 1] 0

Linear-188 [-1, 1280] 1,230,080

Hardswish-189 [-1, 1280] 0

Dropout-190 [-1, 1280] 0

Linear-191 [-1, 101] 129,381

================================================================

Total params: 4,331,413

Trainable params: 129,381

Non-trainable params: 4,202,032

----------------------------------------------------------------

Input size (MB): 0.57

Forward/backward pass size (MB): 105.41

Params size (MB): 16.52

Estimated Total Size (MB): 122.50

----------------------------------------------------------------

1. Nasnet model for training summary:

----------------------------------------------------------------

Layer (type) Output Shape Param #

================================================================

Conv2d-1 [-1, 96, 111, 111] 2,592

Identity-2 [-1, 96, 111, 111] 0

Identity-3 [-1, 96, 111, 111] 0

BatchNormAct2d-4 [-1, 96, 111, 111] 192

ConvNormAct-5 [-1, 96, 111, 111] 0

ReLU-6 [-1, 96, 111, 111] 0

Conv2d-7 [-1, 42, 111, 111] 4,032

BatchNorm2d-8 [-1, 42, 111, 111] 84

ActConvBn-9 [-1, 42, 111, 111] 0

ReLU-10 [-1, 42, 111, 111] 0

Conv2dSame-11 [-1, 42, 56, 56] 1,050

Conv2d-12 [-1, 42, 56, 56] 1,764

SeparableConv2d-13 [-1, 42, 56, 56] 0

BatchNorm2d-14 [-1, 42, 56, 56] 84

ReLU-15 [-1, 42, 56, 56] 0

Conv2d-16 [-1, 42, 56, 56] 1,050

Conv2d-17 [-1, 42, 56, 56] 1,764

SeparableConv2d-18 [-1, 42, 56, 56] 0

BatchNorm2d-19 [-1, 42, 56, 56] 84

BranchSeparables-20 [-1, 42, 56, 56] 0

ReLU-21 [-1, 96, 111, 111] 0

Conv2dSame-22 [-1, 96, 56, 56] 4,704

Conv2d-23 [-1, 42, 56, 56] 4,032

SeparableConv2d-24 [-1, 42, 56, 56] 0

BatchNorm2d-25 [-1, 42, 56, 56] 84

ReLU-26 [-1, 42, 56, 56] 0

Conv2d-27 [-1, 42, 56, 56] 2,058

Conv2d-28 [-1, 42, 56, 56] 1,764

SeparableConv2d-29 [-1, 42, 56, 56] 0

BatchNorm2d-30 [-1, 42, 56, 56] 84

BranchSeparables-31 [-1, 42, 56, 56] 0

MaxPool2dSame-32 [-1, 42, 56, 56] 0

ReLU-33 [-1, 96, 111, 111] 0

Conv2dSame-34 [-1, 96, 56, 56] 4,704

Conv2d-35 [-1, 42, 56, 56] 4,032

SeparableConv2d-36 [-1, 42, 56, 56] 0

BatchNorm2d-37 [-1, 42, 56, 56] 84

ReLU-38 [-1, 42, 56, 56] 0

Conv2d-39 [-1, 42, 56, 56] 2,058

Conv2d-40 [-1, 42, 56, 56] 1,764

SeparableConv2d-41 [-1, 42, 56, 56] 0

BatchNorm2d-42 [-1, 42, 56, 56] 84

BranchSeparables-43 [-1, 42, 56, 56] 0

AvgPool2dSame-44 [-1, 42, 56, 56] 0

ReLU-45 [-1, 96, 111, 111] 0

Conv2dSame-46 [-1, 96, 56, 56] 2,400

Conv2d-47 [-1, 42, 56, 56] 4,032

SeparableConv2d-48 [-1, 42, 56, 56] 0

BatchNorm2d-49 [-1, 42, 56, 56] 84

ReLU-50 [-1, 42, 56, 56] 0

Conv2d-51 [-1, 42, 56, 56] 1,050

Conv2d-52 [-1, 42, 56, 56] 1,764

SeparableConv2d-53 [-1, 42, 56, 56] 0

BatchNorm2d-54 [-1, 42, 56, 56] 84

BranchSeparables-55 [-1, 42, 56, 56] 0

AvgPool2d-56 [-1, 42, 56, 56] 0

ReLU-57 [-1, 42, 56, 56] 0

Conv2d-58 [-1, 42, 56, 56] 378

Conv2d-59 [-1, 42, 56, 56] 1,764

SeparableConv2d-60 [-1, 42, 56, 56] 0

BatchNorm2d-61 [-1, 42, 56, 56] 84

ReLU-62 [-1, 42, 56, 56] 0

Conv2d-63 [-1, 42, 56, 56] 378

Conv2d-64 [-1, 42, 56, 56] 1,764

SeparableConv2d-65 [-1, 42, 56, 56] 0

BatchNorm2d-66 [-1, 42, 56, 56] 84

BranchSeparables-67 [-1, 42, 56, 56] 0

MaxPool2dSame-68 [-1, 42, 56, 56] 0

CellStem0-69 [-1, 168, 56, 56] 0

ReLU-70 [-1, 168, 56, 56] 0

Conv2d-71 [-1, 84, 56, 56] 14,112

BatchNorm2d-72 [-1, 84, 56, 56] 168

ActConvBn-73 [-1, 84, 56, 56] 0

ReLU-74 [-1, 96, 111, 111] 0

AvgPool2d-75 [-1, 96, 56, 56] 0

Conv2d-76 [-1, 42, 56, 56] 4,032

ZeroPad2d-77 [-1, 96, 111, 111] 0

AvgPool2d-78 [-1, 96, 56, 56] 0

Conv2d-79 [-1, 42, 56, 56] 4,032

BatchNorm2d-80 [-1, 84, 56, 56] 168

ReLU-81 [-1, 84, 56, 56] 0

Conv2dSame-82 [-1, 84, 28, 28] 2,100

Conv2d-83 [-1, 84, 28, 28] 7,056

SeparableConv2d-84 [-1, 84, 28, 28] 0

BatchNorm2d-85 [-1, 84, 28, 28] 168

ReLU-86 [-1, 84, 28, 28] 0

Conv2d-87 [-1, 84, 28, 28] 2,100

Conv2d-88 [-1, 84, 28, 28] 7,056

SeparableConv2d-89 [-1, 84, 28, 28] 0

BatchNorm2d-90 [-1, 84, 28, 28] 168

BranchSeparables-91 [-1, 84, 28, 28] 0

ReLU-92 [-1, 84, 56, 56] 0

Conv2dSame-93 [-1, 84, 28, 28] 4,116

Conv2d-94 [-1, 84, 28, 28] 7,056

SeparableConv2d-95 [-1, 84, 28, 28] 0

BatchNorm2d-96 [-1, 84, 28, 28] 168

ReLU-97 [-1, 84, 28, 28] 0

Conv2d-98 [-1, 84, 28, 28] 4,116

Conv2d-99 [-1, 84, 28, 28] 7,056

SeparableConv2d-100 [-1, 84, 28, 28] 0

BatchNorm2d-101 [-1, 84, 28, 28] 168

BranchSeparables-102 [-1, 84, 28, 28] 0

MaxPool2dSame-103 [-1, 84, 28, 28] 0

ReLU-104 [-1, 84, 56, 56] 0

Conv2dSame-105 [-1, 84, 28, 28] 4,116

Conv2d-106 [-1, 84, 28, 28] 7,056

SeparableConv2d-107 [-1, 84, 28, 28] 0

BatchNorm2d-108 [-1, 84, 28, 28] 168

ReLU-109 [-1, 84, 28, 28] 0

Conv2d-110 [-1, 84, 28, 28] 4,116

Conv2d-111 [-1, 84, 28, 28] 7,056

SeparableConv2d-112 [-1, 84, 28, 28] 0

BatchNorm2d-113 [-1, 84, 28, 28] 168

BranchSeparables-114 [-1, 84, 28, 28] 0

AvgPool2dSame-115 [-1, 84, 28, 28] 0

ReLU-116 [-1, 84, 56, 56] 0

Conv2dSame-117 [-1, 84, 28, 28] 2,100

Conv2d-118 [-1, 84, 28, 28] 7,056

SeparableConv2d-119 [-1, 84, 28, 28] 0

BatchNorm2d-120 [-1, 84, 28, 28] 168

ReLU-121 [-1, 84, 28, 28] 0

Conv2d-122 [-1, 84, 28, 28] 2,100

Conv2d-123 [-1, 84, 28, 28] 7,056

SeparableConv2d-124 [-1, 84, 28, 28] 0

BatchNorm2d-125 [-1, 84, 28, 28] 168

BranchSeparables-126 [-1, 84, 28, 28] 0

AvgPool2d-127 [-1, 84, 28, 28] 0

ReLU-128 [-1, 84, 28, 28] 0

Conv2d-129 [-1, 84, 28, 28] 756

Conv2d-130 [-1, 84, 28, 28] 7,056

SeparableConv2d-131 [-1, 84, 28, 28] 0

BatchNorm2d-132 [-1, 84, 28, 28] 168

ReLU-133 [-1, 84, 28, 28] 0

Conv2d-134 [-1, 84, 28, 28] 756

Conv2d-135 [-1, 84, 28, 28] 7,056

SeparableConv2d-136 [-1, 84, 28, 28] 0

BatchNorm2d-137 [-1, 84, 28, 28] 168

BranchSeparables-138 [-1, 84, 28, 28] 0

MaxPool2dSame-139 [-1, 84, 28, 28] 0

CellStem1-140 [-1, 336, 28, 28] 0

ReLU-141 [-1, 168, 56, 56] 0

AvgPool2d-142 [-1, 168, 28, 28] 0

Conv2d-143 [-1, 84, 28, 28] 14,112

ZeroPad2d-144 [-1, 168, 56, 56] 0

AvgPool2d-145 [-1, 168, 28, 28] 0

Conv2d-146 [-1, 84, 28, 28] 14,112

BatchNorm2d-147 [-1, 168, 28, 28] 336

ReLU-148 [-1, 336, 28, 28] 0

Conv2d-149 [-1, 168, 28, 28] 56,448

BatchNorm2d-150 [-1, 168, 28, 28] 336

ActConvBn-151 [-1, 168, 28, 28] 0

ReLU-152 [-1, 168, 28, 28] 0

Conv2d-153 [-1, 168, 28, 28] 4,200

Conv2d-154 [-1, 168, 28, 28] 28,224

SeparableConv2d-155 [-1, 168, 28, 28] 0

BatchNorm2d-156 [-1, 168, 28, 28] 336

ReLU-157 [-1, 168, 28, 28] 0

Conv2d-158 [-1, 168, 28, 28] 4,200

Conv2d-159 [-1, 168, 28, 28] 28,224

SeparableConv2d-160 [-1, 168, 28, 28] 0

BatchNorm2d-161 [-1, 168, 28, 28] 336

BranchSeparables-162 [-1, 168, 28, 28] 0

ReLU-163 [-1, 168, 28, 28] 0

Conv2d-164 [-1, 168, 28, 28] 1,512

Conv2d-165 [-1, 168, 28, 28] 28,224

SeparableConv2d-166 [-1, 168, 28, 28] 0

BatchNorm2d-167 [-1, 168, 28, 28] 336

ReLU-168 [-1, 168, 28, 28] 0

Conv2d-169 [-1, 168, 28, 28] 1,512

Conv2d-170 [-1, 168, 28, 28] 28,224

SeparableConv2d-171 [-1, 168, 28, 28] 0

BatchNorm2d-172 [-1, 168, 28, 28] 336

BranchSeparables-173 [-1, 168, 28, 28] 0

ReLU-174 [-1, 168, 28, 28] 0

Conv2d-175 [-1, 168, 28, 28] 4,200

Conv2d-176 [-1, 168, 28, 28] 28,224

SeparableConv2d-177 [-1, 168, 28, 28] 0

BatchNorm2d-178 [-1, 168, 28, 28] 336

ReLU-179 [-1, 168, 28, 28] 0

Conv2d-180 [-1, 168, 28, 28] 4,200

Conv2d-181 [-1, 168, 28, 28] 28,224

SeparableConv2d-182 [-1, 168, 28, 28] 0

BatchNorm2d-183 [-1, 168, 28, 28] 336

BranchSeparables-184 [-1, 168, 28, 28] 0

ReLU-185 [-1, 168, 28, 28] 0

Conv2d-186 [-1, 168, 28, 28] 1,512

Conv2d-187 [-1, 168, 28, 28] 28,224

SeparableConv2d-188 [-1, 168, 28, 28] 0

BatchNorm2d-189 [-1, 168, 28, 28] 336

ReLU-190 [-1, 168, 28, 28] 0

Conv2d-191 [-1, 168, 28, 28] 1,512

Conv2d-192 [-1, 168, 28, 28] 28,224

SeparableConv2d-193 [-1, 168, 28, 28] 0

BatchNorm2d-194 [-1, 168, 28, 28] 336

BranchSeparables-195 [-1, 168, 28, 28] 0

AvgPool2d-196 [-1, 168, 28, 28] 0

AvgPool2d-197 [-1, 168, 28, 28] 0

AvgPool2d-198 [-1, 168, 28, 28] 0

ReLU-199 [-1, 168, 28, 28] 0

Conv2d-200 [-1, 168, 28, 28] 1,512

Conv2d-201 [-1, 168, 28, 28] 28,224

SeparableConv2d-202 [-1, 168, 28, 28] 0

BatchNorm2d-203 [-1, 168, 28, 28] 336

ReLU-204 [-1, 168, 28, 28] 0

Conv2d-205 [-1, 168, 28, 28] 1,512

Conv2d-206 [-1, 168, 28, 28] 28,224

SeparableConv2d-207 [-1, 168, 28, 28] 0

BatchNorm2d-208 [-1, 168, 28, 28] 336

BranchSeparables-209 [-1, 168, 28, 28] 0

FirstCell-210 [-1, 1008, 28, 28] 0

ReLU-211 [-1, 336, 28, 28] 0

Conv2d-212 [-1, 168, 28, 28] 56,448

BatchNorm2d-213 [-1, 168, 28, 28] 336

ActConvBn-214 [-1, 168, 28, 28] 0

ReLU-215 [-1, 1008, 28, 28] 0

Conv2d-216 [-1, 168, 28, 28] 169,344

BatchNorm2d-217 [-1, 168, 28, 28] 336

ActConvBn-218 [-1, 168, 28, 28] 0

ReLU-219 [-1, 168, 28, 28] 0

Conv2d-220 [-1, 168, 28, 28] 4,200

Conv2d-221 [-1, 168, 28, 28] 28,224

SeparableConv2d-222 [-1, 168, 28, 28] 0

BatchNorm2d-223 [-1, 168, 28, 28] 336

ReLU-224 [-1, 168, 28, 28] 0

Conv2d-225 [-1, 168, 28, 28] 4,200

Conv2d-226 [-1, 168, 28, 28] 28,224

SeparableConv2d-227 [-1, 168, 28, 28] 0

BatchNorm2d-228 [-1, 168, 28, 28] 336

BranchSeparables-229 [-1, 168, 28, 28] 0

ReLU-230 [-1, 168, 28, 28] 0

Conv2d-231 [-1, 168, 28, 28] 1,512

Conv2d-232 [-1, 168, 28, 28] 28,224

SeparableConv2d-233 [-1, 168, 28, 28] 0

BatchNorm2d-234 [-1, 168, 28, 28] 336

ReLU-235 [-1, 168, 28, 28] 0

Conv2d-236 [-1, 168, 28, 28] 1,512

Conv2d-237 [-1, 168, 28, 28] 28,224

SeparableConv2d-238 [-1, 168, 28, 28] 0

BatchNorm2d-239 [-1, 168, 28, 28] 336

BranchSeparables-240 [-1, 168, 28, 28] 0

ReLU-241 [-1, 168, 28, 28] 0

Conv2d-242 [-1, 168, 28, 28] 4,200

Conv2d-243 [-1, 168, 28, 28] 28,224

SeparableConv2d-244 [-1, 168, 28, 28] 0

BatchNorm2d-245 [-1, 168, 28, 28] 336

ReLU-246 [-1, 168, 28, 28] 0

Conv2d-247 [-1, 168, 28, 28] 4,200

Conv2d-248 [-1, 168, 28, 28] 28,224

SeparableConv2d-249 [-1, 168, 28, 28] 0

BatchNorm2d-250 [-1, 168, 28, 28] 336

BranchSeparables-251 [-1, 168, 28, 28] 0

ReLU-252 [-1, 168, 28, 28] 0

Conv2d-253 [-1, 168, 28, 28] 1,512

Conv2d-254 [-1, 168, 28, 28] 28,224

SeparableConv2d-255 [-1, 168, 28, 28] 0

BatchNorm2d-256 [-1, 168, 28, 28] 336

ReLU-257 [-1, 168, 28, 28] 0

Conv2d-258 [-1, 168, 28, 28] 1,512

Conv2d-259 [-1, 168, 28, 28] 28,224

SeparableConv2d-260 [-1, 168, 28, 28] 0

BatchNorm2d-261 [-1, 168, 28, 28] 336

BranchSeparables-262 [-1, 168, 28, 28] 0

AvgPool2d-263 [-1, 168, 28, 28] 0

AvgPool2d-264 [-1, 168, 28, 28] 0

AvgPool2d-265 [-1, 168, 28, 28] 0

ReLU-266 [-1, 168, 28, 28] 0

Conv2d-267 [-1, 168, 28, 28] 1,512

Conv2d-268 [-1, 168, 28, 28] 28,224

SeparableConv2d-269 [-1, 168, 28, 28] 0

BatchNorm2d-270 [-1, 168, 28, 28] 336

ReLU-271 [-1, 168, 28, 28] 0

Conv2d-272 [-1, 168, 28, 28] 1,512

Conv2d-273 [-1, 168, 28, 28] 28,224

SeparableConv2d-274 [-1, 168, 28, 28] 0

BatchNorm2d-275 [-1, 168, 28, 28] 336

BranchSeparables-276 [-1, 168, 28, 28] 0

NormalCell-277 [-1, 1008, 28, 28] 0

ReLU-278 [-1, 1008, 28, 28] 0

Conv2d-279 [-1, 168, 28, 28] 169,344

BatchNorm2d-280 [-1, 168, 28, 28] 336

ActConvBn-281 [-1, 168, 28, 28] 0

ReLU-282 [-1, 1008, 28, 28] 0

Conv2d-283 [-1, 168, 28, 28] 169,344

BatchNorm2d-284 [-1, 168, 28, 28] 336

ActConvBn-285 [-1, 168, 28, 28] 0

ReLU-286 [-1, 168, 28, 28] 0

Conv2d-287 [-1, 168, 28, 28] 4,200

Conv2d-288 [-1, 168, 28, 28] 28,224

SeparableConv2d-289 [-1, 168, 28, 28] 0

BatchNorm2d-290 [-1, 168, 28, 28] 336

ReLU-291 [-1, 168, 28, 28] 0

Conv2d-292 [-1, 168, 28, 28] 4,200

Conv2d-293 [-1, 168, 28, 28] 28,224

SeparableConv2d-294 [-1, 168, 28, 28] 0

BatchNorm2d-295 [-1, 168, 28, 28] 336

BranchSeparables-296 [-1, 168, 28, 28] 0

ReLU-297 [-1, 168, 28, 28] 0

Conv2d-298 [-1, 168, 28, 28] 1,512

Conv2d-299 [-1, 168, 28, 28] 28,224

SeparableConv2d-300 [-1, 168, 28, 28] 0

BatchNorm2d-301 [-1, 168, 28, 28] 336

ReLU-302 [-1, 168, 28, 28] 0

Conv2d-303 [-1, 168, 28, 28] 1,512

Conv2d-304 [-1, 168, 28, 28] 28,224

SeparableConv2d-305 [-1, 168, 28, 28] 0

BatchNorm2d-306 [-1, 168, 28, 28] 336

BranchSeparables-307 [-1, 168, 28, 28] 0

ReLU-308 [-1, 168, 28, 28] 0

Conv2d-309 [-1, 168, 28, 28] 4,200

Conv2d-310 [-1, 168, 28, 28] 28,224

SeparableConv2d-311 [-1, 168, 28, 28] 0

BatchNorm2d-312 [-1, 168, 28, 28] 336

ReLU-313 [-1, 168, 28, 28] 0

Conv2d-314 [-1, 168, 28, 28] 4,200

Conv2d-315 [-1, 168, 28, 28] 28,224

SeparableConv2d-316 [-1, 168, 28, 28] 0

BatchNorm2d-317 [-1, 168, 28, 28] 336

BranchSeparables-318 [-1, 168, 28, 28] 0

ReLU-319 [-1, 168, 28, 28] 0

Conv2d-320 [-1, 168, 28, 28] 1,512

Conv2d-321 [-1, 168, 28, 28] 28,224

SeparableConv2d-322 [-1, 168, 28, 28] 0

BatchNorm2d-323 [-1, 168, 28, 28] 336

ReLU-324 [-1, 168, 28, 28] 0

Conv2d-325 [-1, 168, 28, 28] 1,512

Conv2d-326 [-1, 168, 28, 28] 28,224

SeparableConv2d-327 [-1, 168, 28, 28] 0

BatchNorm2d-328 [-1, 168, 28, 28] 336

BranchSeparables-329 [-1, 168, 28, 28] 0

AvgPool2d-330 [-1, 168, 28, 28] 0

AvgPool2d-331 [-1, 168, 28, 28] 0

AvgPool2d-332 [-1, 168, 28, 28] 0

ReLU-333 [-1, 168, 28, 28] 0

Conv2d-334 [-1, 168, 28, 28] 1,512

Conv2d-335 [-1, 168, 28, 28] 28,224

SeparableConv2d-336 [-1, 168, 28, 28] 0

BatchNorm2d-337 [-1, 168, 28, 28] 336

ReLU-338 [-1, 168, 28, 28] 0

Conv2d-339 [-1, 168, 28, 28] 1,512

Conv2d-340 [-1, 168, 28, 28] 28,224

SeparableConv2d-341 [-1, 168, 28, 28] 0

BatchNorm2d-342 [-1, 168, 28, 28] 336

BranchSeparables-343 [-1, 168, 28, 28] 0

NormalCell-344 [-1, 1008, 28, 28] 0

ReLU-345 [-1, 1008, 28, 28] 0

Conv2d-346 [-1, 168, 28, 28] 169,344

BatchNorm2d-347 [-1, 168, 28, 28] 336

ActConvBn-348 [-1, 168, 28, 28] 0

ReLU-349 [-1, 1008, 28, 28] 0

Conv2d-350 [-1, 168, 28, 28] 169,344

BatchNorm2d-351 [-1, 168, 28, 28] 336

ActConvBn-352 [-1, 168, 28, 28] 0

ReLU-353 [-1, 168, 28, 28] 0

Conv2d-354 [-1, 168, 28, 28] 4,200

Conv2d-355 [-1, 168, 28, 28] 28,224

SeparableConv2d-356 [-1, 168, 28, 28] 0

BatchNorm2d-357 [-1, 168, 28, 28] 336

ReLU-358 [-1, 168, 28, 28] 0

Conv2d-359 [-1, 168, 28, 28] 4,200

Conv2d-360 [-1, 168, 28, 28] 28,224

SeparableConv2d-361 [-1, 168, 28, 28] 0

BatchNorm2d-362 [-1, 168, 28, 28] 336

BranchSeparables-363 [-1, 168, 28, 28] 0

ReLU-364 [-1, 168, 28, 28] 0

Conv2d-365 [-1, 168, 28, 28] 1,512

Conv2d-366 [-1, 168, 28, 28] 28,224

SeparableConv2d-367 [-1, 168, 28, 28] 0

BatchNorm2d-368 [-1, 168, 28, 28] 336

ReLU-369 [-1, 168, 28, 28] 0

Conv2d-370 [-1, 168, 28, 28] 1,512

Conv2d-371 [-1, 168, 28, 28] 28,224

SeparableConv2d-372 [-1, 168, 28, 28] 0

BatchNorm2d-373 [-1, 168, 28, 28] 336

BranchSeparables-374 [-1, 168, 28, 28] 0

ReLU-375 [-1, 168, 28, 28] 0

Conv2d-376 [-1, 168, 28, 28] 4,200

Conv2d-377 [-1, 168, 28, 28] 28,224

SeparableConv2d-378 [-1, 168, 28, 28] 0

BatchNorm2d-379 [-1, 168, 28, 28] 336

ReLU-380 [-1, 168, 28, 28] 0

Conv2d-381 [-1, 168, 28, 28] 4,200

Conv2d-382 [-1, 168, 28, 28] 28,224

SeparableConv2d-383 [-1, 168, 28, 28] 0

BatchNorm2d-384 [-1, 168, 28, 28] 336

BranchSeparables-385 [-1, 168, 28, 28] 0

ReLU-386 [-1, 168, 28, 28] 0

Conv2d-387 [-1, 168, 28, 28] 1,512

Conv2d-388 [-1, 168, 28, 28] 28,224

SeparableConv2d-389 [-1, 168, 28, 28] 0

BatchNorm2d-390 [-1, 168, 28, 28] 336

ReLU-391 [-1, 168, 28, 28] 0

Conv2d-392 [-1, 168, 28, 28] 1,512

Conv2d-393 [-1, 168, 28, 28] 28,224

SeparableConv2d-394 [-1, 168, 28, 28] 0

BatchNorm2d-395 [-1, 168, 28, 28] 336

BranchSeparables-396 [-1, 168, 28, 28] 0

AvgPool2d-397 [-1, 168, 28, 28] 0

AvgPool2d-398 [-1, 168, 28, 28] 0

AvgPool2d-399 [-1, 168, 28, 28] 0

ReLU-400 [-1, 168, 28, 28] 0

Conv2d-401 [-1, 168, 28, 28] 1,512

Conv2d-402 [-1, 168, 28, 28] 28,224

SeparableConv2d-403 [-1, 168, 28, 28] 0

BatchNorm2d-404 [-1, 168, 28, 28] 336

ReLU-405 [-1, 168, 28, 28] 0

Conv2d-406 [-1, 168, 28, 28] 1,512

Conv2d-407 [-1, 168, 28, 28] 28,224

SeparableConv2d-408 [-1, 168, 28, 28] 0

BatchNorm2d-409 [-1, 168, 28, 28] 336

BranchSeparables-410 [-1, 168, 28, 28] 0

NormalCell-411 [-1, 1008, 28, 28] 0

ReLU-412 [-1, 1008, 28, 28] 0

Conv2d-413 [-1, 168, 28, 28] 169,344

BatchNorm2d-414 [-1, 168, 28, 28] 336

ActConvBn-415 [-1, 168, 28, 28] 0

ReLU-416 [-1, 1008, 28, 28] 0

Conv2d-417 [-1, 168, 28, 28] 169,344

BatchNorm2d-418 [-1, 168, 28, 28] 336

ActConvBn-419 [-1, 168, 28, 28] 0

ReLU-420 [-1, 168, 28, 28] 0

Conv2d-421 [-1, 168, 28, 28] 4,200

Conv2d-422 [-1, 168, 28, 28] 28,224

SeparableConv2d-423 [-1, 168, 28, 28] 0

BatchNorm2d-424 [-1, 168, 28, 28] 336

ReLU-425 [-1, 168, 28, 28] 0

Conv2d-426 [-1, 168, 28, 28] 4,200

Conv2d-427 [-1, 168, 28, 28] 28,224

SeparableConv2d-428 [-1, 168, 28, 28] 0

BatchNorm2d-429 [-1, 168, 28, 28] 336

BranchSeparables-430 [-1, 168, 28, 28] 0

ReLU-431 [-1, 168, 28, 28] 0

Conv2d-432 [-1, 168, 28, 28] 1,512

Conv2d-433 [-1, 168, 28, 28] 28,224

SeparableConv2d-434 [-1, 168, 28, 28] 0

BatchNorm2d-435 [-1, 168, 28, 28] 336

ReLU-436 [-1, 168, 28, 28] 0

Conv2d-437 [-1, 168, 28, 28] 1,512

Conv2d-438 [-1, 168, 28, 28] 28,224

SeparableConv2d-439 [-1, 168, 28, 28] 0

BatchNorm2d-440 [-1, 168, 28, 28] 336

BranchSeparables-441 [-1, 168, 28, 28] 0

ReLU-442 [-1, 168, 28, 28] 0

Conv2d-443 [-1, 168, 28, 28] 4,200

Conv2d-444 [-1, 168, 28, 28] 28,224

SeparableConv2d-445 [-1, 168, 28, 28] 0

BatchNorm2d-446 [-1, 168, 28, 28] 336

ReLU-447 [-1, 168, 28, 28] 0

Conv2d-448 [-1, 168, 28, 28] 4,200

Conv2d-449 [-1, 168, 28, 28] 28,224

SeparableConv2d-450 [-1, 168, 28, 28] 0

BatchNorm2d-451 [-1, 168, 28, 28] 336

BranchSeparables-452 [-1, 168, 28, 28] 0

ReLU-453 [-1, 168, 28, 28] 0

Conv2d-454 [-1, 168, 28, 28] 1,512

Conv2d-455 [-1, 168, 28, 28] 28,224

SeparableConv2d-456 [-1, 168, 28, 28] 0

BatchNorm2d-457 [-1, 168, 28, 28] 336

ReLU-458 [-1, 168, 28, 28] 0

Conv2d-459 [-1, 168, 28, 28] 1,512

Conv2d-460 [-1, 168, 28, 28] 28,224

SeparableConv2d-461 [-1, 168, 28, 28] 0

BatchNorm2d-462 [-1, 168, 28, 28] 336

BranchSeparables-463 [-1, 168, 28, 28] 0

AvgPool2d-464 [-1, 168, 28, 28] 0

AvgPool2d-465 [-1, 168, 28, 28] 0

AvgPool2d-466 [-1, 168, 28, 28] 0

ReLU-467 [-1, 168, 28, 28] 0

Conv2d-468 [-1, 168, 28, 28] 1,512

Conv2d-469 [-1, 168, 28, 28] 28,224

SeparableConv2d-470 [-1, 168, 28, 28] 0

BatchNorm2d-471 [-1, 168, 28, 28] 336

ReLU-472 [-1, 168, 28, 28] 0

Conv2d-473 [-1, 168, 28, 28] 1,512

Conv2d-474 [-1, 168, 28, 28] 28,224

SeparableConv2d-475 [-1, 168, 28, 28] 0

BatchNorm2d-476 [-1, 168, 28, 28] 336

BranchSeparables-477 [-1, 168, 28, 28] 0

NormalCell-478 [-1, 1008, 28, 28] 0

ReLU-479 [-1, 1008, 28, 28] 0

Conv2d-480 [-1, 168, 28, 28] 169,344

BatchNorm2d-481 [-1, 168, 28, 28] 336

ActConvBn-482 [-1, 168, 28, 28] 0

ReLU-483 [-1, 1008, 28, 28] 0

Conv2d-484 [-1, 168, 28, 28] 169,344

BatchNorm2d-485 [-1, 168, 28, 28] 336

ActConvBn-486 [-1, 168, 28, 28] 0

ReLU-487 [-1, 168, 28, 28] 0

Conv2d-488 [-1, 168, 28, 28] 4,200

Conv2d-489 [-1, 168, 28, 28] 28,224

SeparableConv2d-490 [-1, 168, 28, 28] 0

BatchNorm2d-491 [-1, 168, 28, 28] 336

ReLU-492 [-1, 168, 28, 28] 0

Conv2d-493 [-1, 168, 28, 28] 4,200

Conv2d-494 [-1, 168, 28, 28] 28,224

SeparableConv2d-495 [-1, 168, 28, 28] 0

BatchNorm2d-496 [-1, 168, 28, 28] 336

BranchSeparables-497 [-1, 168, 28, 28] 0

ReLU-498 [-1, 168, 28, 28] 0

Conv2d-499 [-1, 168, 28, 28] 1,512

Conv2d-500 [-1, 168, 28, 28] 28,224

SeparableConv2d-501 [-1, 168, 28, 28] 0

BatchNorm2d-502 [-1, 168, 28, 28] 336

ReLU-503 [-1, 168, 28, 28] 0

Conv2d-504 [-1, 168, 28, 28] 1,512

Conv2d-505 [-1, 168, 28, 28] 28,224

SeparableConv2d-506 [-1, 168, 28, 28] 0

BatchNorm2d-507 [-1, 168, 28, 28] 336

BranchSeparables-508 [-1, 168, 28, 28] 0

ReLU-509 [-1, 168, 28, 28] 0

Conv2d-510 [-1, 168, 28, 28] 4,200

Conv2d-511 [-1, 168, 28, 28] 28,224

SeparableConv2d-512 [-1, 168, 28, 28] 0

BatchNorm2d-513 [-1, 168, 28, 28] 336

ReLU-514 [-1, 168, 28, 28] 0

Conv2d-515 [-1, 168, 28, 28] 4,200

Conv2d-516 [-1, 168, 28, 28] 28,224

SeparableConv2d-517 [-1, 168, 28, 28] 0

BatchNorm2d-518 [-1, 168, 28, 28] 336

BranchSeparables-519 [-1, 168, 28, 28] 0

ReLU-520 [-1, 168, 28, 28] 0

Conv2d-521 [-1, 168, 28, 28] 1,512

Conv2d-522 [-1, 168, 28, 28] 28,224

SeparableConv2d-523 [-1, 168, 28, 28] 0

BatchNorm2d-524 [-1, 168, 28, 28] 336

ReLU-525 [-1, 168, 28, 28] 0

Conv2d-526 [-1, 168, 28, 28] 1,512

Conv2d-527 [-1, 168, 28, 28] 28,224

SeparableConv2d-528 [-1, 168, 28, 28] 0

BatchNorm2d-529 [-1, 168, 28, 28] 336

BranchSeparables-530 [-1, 168, 28, 28] 0

AvgPool2d-531 [-1, 168, 28, 28] 0

AvgPool2d-532 [-1, 168, 28, 28] 0

AvgPool2d-533 [-1, 168, 28, 28] 0

ReLU-534 [-1, 168, 28, 28] 0

Conv2d-535 [-1, 168, 28, 28] 1,512

Conv2d-536 [-1, 168, 28, 28] 28,224

SeparableConv2d-537 [-1, 168, 28, 28] 0

BatchNorm2d-538 [-1, 168, 28, 28] 336

ReLU-539 [-1, 168, 28, 28] 0

Conv2d-540 [-1, 168, 28, 28] 1,512

Conv2d-541 [-1, 168, 28, 28] 28,224

SeparableConv2d-542 [-1, 168, 28, 28] 0

BatchNorm2d-543 [-1, 168, 28, 28] 336

BranchSeparables-544 [-1, 168, 28, 28] 0

NormalCell-545 [-1, 1008, 28, 28] 0

ReLU-546 [-1, 1008, 28, 28] 0

Conv2d-547 [-1, 336, 28, 28] 338,688

BatchNorm2d-548 [-1, 336, 28, 28] 672

ActConvBn-549 [-1, 336, 28, 28] 0

ReLU-550 [-1, 1008, 28, 28] 0

Conv2d-551 [-1, 336, 28, 28] 338,688

BatchNorm2d-552 [-1, 336, 28, 28] 672

ActConvBn-553 [-1, 336, 28, 28] 0

ReLU-554 [-1, 336, 28, 28] 0

Conv2dSame-555 [-1, 336, 14, 14] 8,400

Conv2d-556 [-1, 336, 14, 14] 112,896

SeparableConv2d-557 [-1, 336, 14, 14] 0

BatchNorm2d-558 [-1, 336, 14, 14] 672

ReLU-559 [-1, 336, 14, 14] 0

Conv2d-560 [-1, 336, 14, 14] 8,400

Conv2d-561 [-1, 336, 14, 14] 112,896

SeparableConv2d-562 [-1, 336, 14, 14] 0

BatchNorm2d-563 [-1, 336, 14, 14] 672

BranchSeparables-564 [-1, 336, 14, 14] 0

ReLU-565 [-1, 336, 28, 28] 0

Conv2dSame-566 [-1, 336, 14, 14] 16,464

Conv2d-567 [-1, 336, 14, 14] 112,896

SeparableConv2d-568 [-1, 336, 14, 14] 0

BatchNorm2d-569 [-1, 336, 14, 14] 672

ReLU-570 [-1, 336, 14, 14] 0

Conv2d-571 [-1, 336, 14, 14] 16,464

Conv2d-572 [-1, 336, 14, 14] 112,896

SeparableConv2d-573 [-1, 336, 14, 14] 0

BatchNorm2d-574 [-1, 336, 14, 14] 672

BranchSeparables-575 [-1, 336, 14, 14] 0

MaxPool2dSame-576 [-1, 336, 14, 14] 0

ReLU-577 [-1, 336, 28, 28] 0

Conv2dSame-578 [-1, 336, 14, 14] 16,464

Conv2d-579 [-1, 336, 14, 14] 112,896

SeparableConv2d-580 [-1, 336, 14, 14] 0

BatchNorm2d-581 [-1, 336, 14, 14] 672

ReLU-582 [-1, 336, 14, 14] 0

Conv2d-583 [-1, 336, 14, 14] 16,464

Conv2d-584 [-1, 336, 14, 14] 112,896

SeparableConv2d-585 [-1, 336, 14, 14] 0

BatchNorm2d-586 [-1, 336, 14, 14] 672

BranchSeparables-587 [-1, 336, 14, 14] 0

AvgPool2dSame-588 [-1, 336, 14, 14] 0

ReLU-589 [-1, 336, 28, 28] 0

Conv2dSame-590 [-1, 336, 14, 14] 8,400

Conv2d-591 [-1, 336, 14, 14] 112,896

SeparableConv2d-592 [-1, 336, 14, 14] 0

BatchNorm2d-593 [-1, 336, 14, 14] 672

ReLU-594 [-1, 336, 14, 14] 0

Conv2d-595 [-1, 336, 14, 14] 8,400

Conv2d-596 [-1, 336, 14, 14] 112,896

SeparableConv2d-597 [-1, 336, 14, 14] 0

BatchNorm2d-598 [-1, 336, 14, 14] 672

BranchSeparables-599 [-1, 336, 14, 14] 0

AvgPool2d-600 [-1, 336, 14, 14] 0

ReLU-601 [-1, 336, 14, 14] 0

Conv2d-602 [-1, 336, 14, 14] 3,024

Conv2d-603 [-1, 336, 14, 14] 112,896

SeparableConv2d-604 [-1, 336, 14, 14] 0

BatchNorm2d-605 [-1, 336, 14, 14] 672

ReLU-606 [-1, 336, 14, 14] 0

Conv2d-607 [-1, 336, 14, 14] 3,024

Conv2d-608 [-1, 336, 14, 14] 112,896

SeparableConv2d-609 [-1, 336, 14, 14] 0

BatchNorm2d-610 [-1, 336, 14, 14] 672

BranchSeparables-611 [-1, 336, 14, 14] 0

MaxPool2dSame-612 [-1, 336, 14, 14] 0

ReductionCell0-613 [-1, 1344, 14, 14] 0

ReLU-614 [-1, 1008, 28, 28] 0

AvgPool2d-615 [-1, 1008, 14, 14] 0

Conv2d-616 [-1, 168, 14, 14] 169,344

ZeroPad2d-617 [-1, 1008, 28, 28] 0

AvgPool2d-618 [-1, 1008, 14, 14] 0

Conv2d-619 [-1, 168, 14, 14] 169,344

BatchNorm2d-620 [-1, 336, 14, 14] 672

ReLU-621 [-1, 1344, 14, 14] 0

Conv2d-622 [-1, 336, 14, 14] 451,584

BatchNorm2d-623 [-1, 336, 14, 14] 672

ActConvBn-624 [-1, 336, 14, 14] 0

ReLU-625 [-1, 336, 14, 14] 0

Conv2d-626 [-1, 336, 14, 14] 8,400

Conv2d-627 [-1, 336, 14, 14] 112,896

SeparableConv2d-628 [-1, 336, 14, 14] 0

BatchNorm2d-629 [-1, 336, 14, 14] 672

ReLU-630 [-1, 336, 14, 14] 0

Conv2d-631 [-1, 336, 14, 14] 8,400

Conv2d-632 [-1, 336, 14, 14] 112,896

SeparableConv2d-633 [-1, 336, 14, 14] 0

BatchNorm2d-634 [-1, 336, 14, 14] 672

BranchSeparables-635 [-1, 336, 14, 14] 0

ReLU-636 [-1, 336, 14, 14] 0

Conv2d-637 [-1, 336, 14, 14] 3,024

Conv2d-638 [-1, 336, 14, 14] 112,896

SeparableConv2d-639 [-1, 336, 14, 14] 0

BatchNorm2d-640 [-1, 336, 14, 14] 672

ReLU-641 [-1, 336, 14, 14] 0

Conv2d-642 [-1, 336, 14, 14] 3,024

Conv2d-643 [-1, 336, 14, 14] 112,896

SeparableConv2d-644 [-1, 336, 14, 14] 0

BatchNorm2d-645 [-1, 336, 14, 14] 672

BranchSeparables-646 [-1, 336, 14, 14] 0

ReLU-647 [-1, 336, 14, 14] 0

Conv2d-648 [-1, 336, 14, 14] 8,400

Conv2d-649 [-1, 336, 14, 14] 112,896

SeparableConv2d-650 [-1, 336, 14, 14] 0

BatchNorm2d-651 [-1, 336, 14, 14] 672

ReLU-652 [-1, 336, 14, 14] 0

Conv2d-653 [-1, 336, 14, 14] 8,400

Conv2d-654 [-1, 336, 14, 14] 112,896

SeparableConv2d-655 [-1, 336, 14, 14] 0

BatchNorm2d-656 [-1, 336, 14, 14] 672

BranchSeparables-657 [-1, 336, 14, 14] 0

ReLU-658 [-1, 336, 14, 14] 0

Conv2d-659 [-1, 336, 14, 14] 3,024

Conv2d-660 [-1, 336, 14, 14] 112,896

SeparableConv2d-661 [-1, 336, 14, 14] 0

BatchNorm2d-662 [-1, 336, 14, 14] 672

ReLU-663 [-1, 336, 14, 14] 0

Conv2d-664 [-1, 336, 14, 14] 3,024

Conv2d-665 [-1, 336, 14, 14] 112,896

SeparableConv2d-666 [-1, 336, 14, 14] 0

BatchNorm2d-667 [-1, 336, 14, 14] 672

BranchSeparables-668 [-1, 336, 14, 14] 0

AvgPool2d-669 [-1, 336, 14, 14] 0

AvgPool2d-670 [-1, 336, 14, 14] 0

AvgPool2d-671 [-1, 336, 14, 14] 0

ReLU-672 [-1, 336, 14, 14] 0

Conv2d-673 [-1, 336, 14, 14] 3,024

Conv2d-674 [-1, 336, 14, 14] 112,896

SeparableConv2d-675 [-1, 336, 14, 14] 0

BatchNorm2d-676 [-1, 336, 14, 14] 672

ReLU-677 [-1, 336, 14, 14] 0

Conv2d-678 [-1, 336, 14, 14] 3,024

Conv2d-679 [-1, 336, 14, 14] 112,896

SeparableConv2d-680 [-1, 336, 14, 14] 0

BatchNorm2d-681 [-1, 336, 14, 14] 672

BranchSeparables-682 [-1, 336, 14, 14] 0

FirstCell-683 [-1, 2016, 14, 14] 0

ReLU-684 [-1, 1344, 14, 14] 0

Conv2d-685 [-1, 336, 14, 14] 451,584

BatchNorm2d-686 [-1, 336, 14, 14] 672

ActConvBn-687 [-1, 336, 14, 14] 0

ReLU-688 [-1, 2016, 14, 14] 0

Conv2d-689 [-1, 336, 14, 14] 677,376

BatchNorm2d-690 [-1, 336, 14, 14] 672

ActConvBn-691 [-1, 336, 14, 14] 0

ReLU-692 [-1, 336, 14, 14] 0

Conv2d-693 [-1, 336, 14, 14] 8,400

Conv2d-694 [-1, 336, 14, 14] 112,896

SeparableConv2d-695 [-1, 336, 14, 14] 0

BatchNorm2d-696 [-1, 336, 14, 14] 672

ReLU-697 [-1, 336, 14, 14] 0

Conv2d-698 [-1, 336, 14, 14] 8,400

Conv2d-699 [-1, 336, 14, 14] 112,896

SeparableConv2d-700 [-1, 336, 14, 14] 0

BatchNorm2d-701 [-1, 336, 14, 14] 672

BranchSeparables-702 [-1, 336, 14, 14] 0

ReLU-703 [-1, 336, 14, 14] 0

Conv2d-704 [-1, 336, 14, 14] 3,024

Conv2d-705 [-1, 336, 14, 14] 112,896

SeparableConv2d-706 [-1, 336, 14, 14] 0

BatchNorm2d-707 [-1, 336, 14, 14] 672

ReLU-708 [-1, 336, 14, 14] 0

Conv2d-709 [-1, 336, 14, 14] 3,024

Conv2d-710 [-1, 336, 14, 14] 112,896

SeparableConv2d-711 [-1, 336, 14, 14] 0

BatchNorm2d-712 [-1, 336, 14, 14] 672

BranchSeparables-713 [-1, 336, 14, 14] 0

ReLU-714 [-1, 336, 14, 14] 0

Conv2d-715 [-1, 336, 14, 14] 8,400

Conv2d-716 [-1, 336, 14, 14] 112,896

SeparableConv2d-717 [-1, 336, 14, 14] 0

BatchNorm2d-718 [-1, 336, 14, 14] 672

ReLU-719 [-1, 336, 14, 14] 0

Conv2d-720 [-1, 336, 14, 14] 8,400

Conv2d-721 [-1, 336, 14, 14] 112,896

SeparableConv2d-722 [-1, 336, 14, 14] 0

BatchNorm2d-723 [-1, 336, 14, 14] 672

BranchSeparables-724 [-1, 336, 14, 14] 0

ReLU-725 [-1, 336, 14, 14] 0

Conv2d-726 [-1, 336, 14, 14] 3,024

Conv2d-727 [-1, 336, 14, 14] 112,896

SeparableConv2d-728 [-1, 336, 14, 14] 0

BatchNorm2d-729 [-1, 336, 14, 14] 672

ReLU-730 [-1, 336, 14, 14] 0

Conv2d-731 [-1, 336, 14, 14] 3,024

Conv2d-732 [-1, 336, 14, 14] 112,896

SeparableConv2d-733 [-1, 336, 14, 14] 0

BatchNorm2d-734 [-1, 336, 14, 14] 672

BranchSeparables-735 [-1, 336, 14, 14] 0

AvgPool2d-736 [-1, 336, 14, 14] 0

AvgPool2d-737 [-1, 336, 14, 14] 0

AvgPool2d-738 [-1, 336, 14, 14] 0

ReLU-739 [-1, 336, 14, 14] 0

Conv2d-740 [-1, 336, 14, 14] 3,024

Conv2d-741 [-1, 336, 14, 14] 112,896

SeparableConv2d-742 [-1, 336, 14, 14] 0

BatchNorm2d-743 [-1, 336, 14, 14] 672

ReLU-744 [-1, 336, 14, 14] 0

Conv2d-745 [-1, 336, 14, 14] 3,024

Conv2d-746 [-1, 336, 14, 14] 112,896

SeparableConv2d-747 [-1, 336, 14, 14] 0

BatchNorm2d-748 [-1, 336, 14, 14] 672

BranchSeparables-749 [-1, 336, 14, 14] 0

NormalCell-750 [-1, 2016, 14, 14] 0

ReLU-751 [-1, 2016, 14, 14] 0

Conv2d-752 [-1, 336, 14, 14] 677,376

BatchNorm2d-753 [-1, 336, 14, 14] 672

ActConvBn-754 [-1, 336, 14, 14] 0

ReLU-755 [-1, 2016, 14, 14] 0

Conv2d-756 [-1, 336, 14, 14] 677,376

BatchNorm2d-757 [-1, 336, 14, 14] 672

ActConvBn-758 [-1, 336, 14, 14] 0

ReLU-759 [-1, 336, 14, 14] 0

Conv2d-760 [-1, 336, 14, 14] 8,400

Conv2d-761 [-1, 336, 14, 14] 112,896

SeparableConv2d-762 [-1, 336, 14, 14] 0

BatchNorm2d-763 [-1, 336, 14, 14] 672

ReLU-764 [-1, 336, 14, 14] 0

Conv2d-765 [-1, 336, 14, 14] 8,400

Conv2d-766 [-1, 336, 14, 14] 112,896

SeparableConv2d-767 [-1, 336, 14, 14] 0

BatchNorm2d-768 [-1, 336, 14, 14] 672

BranchSeparables-769 [-1, 336, 14, 14] 0

ReLU-770 [-1, 336, 14, 14] 0

Conv2d-771 [-1, 336, 14, 14] 3,024

Conv2d-772 [-1, 336, 14, 14] 112,896

SeparableConv2d-773 [-1, 336, 14, 14] 0

BatchNorm2d-774 [-1, 336, 14, 14] 672

ReLU-775 [-1, 336, 14, 14] 0

Conv2d-776 [-1, 336, 14, 14] 3,024

Conv2d-777 [-1, 336, 14, 14] 112,896

SeparableConv2d-778 [-1, 336, 14, 14] 0

BatchNorm2d-779 [-1, 336, 14, 14] 672

BranchSeparables-780 [-1, 336, 14, 14] 0

ReLU-781 [-1, 336, 14, 14] 0

Conv2d-782 [-1, 336, 14, 14] 8,400

Conv2d-783 [-1, 336, 14, 14] 112,896

SeparableConv2d-784 [-1, 336, 14, 14] 0

BatchNorm2d-785 [-1, 336, 14, 14] 672

ReLU-786 [-1, 336, 14, 14] 0

Conv2d-787 [-1, 336, 14, 14] 8,400

Conv2d-788 [-1, 336, 14, 14] 112,896

SeparableConv2d-789 [-1, 336, 14, 14] 0

BatchNorm2d-790 [-1, 336, 14, 14] 672

BranchSeparables-791 [-1, 336, 14, 14] 0

ReLU-792 [-1, 336, 14, 14] 0

Conv2d-793 [-1, 336, 14, 14] 3,024

Conv2d-794 [-1, 336, 14, 14] 112,896

SeparableConv2d-795 [-1, 336, 14, 14] 0

BatchNorm2d-796 [-1, 336, 14, 14] 672

ReLU-797 [-1, 336, 14, 14] 0

Conv2d-798 [-1, 336, 14, 14] 3,024

Conv2d-799 [-1, 336, 14, 14] 112,896

SeparableConv2d-800 [-1, 336, 14, 14] 0

BatchNorm2d-801 [-1, 336, 14, 14] 672

BranchSeparables-802 [-1, 336, 14, 14] 0

AvgPool2d-803 [-1, 336, 14, 14] 0

AvgPool2d-804 [-1, 336, 14, 14] 0

AvgPool2d-805 [-1, 336, 14, 14] 0

ReLU-806 [-1, 336, 14, 14] 0

Conv2d-807 [-1, 336, 14, 14] 3,024

Conv2d-808 [-1, 336, 14, 14] 112,896

SeparableConv2d-809 [-1, 336, 14, 14] 0

BatchNorm2d-810 [-1, 336, 14, 14] 672

ReLU-811 [-1, 336, 14, 14] 0

Conv2d-812 [-1, 336, 14, 14] 3,024

Conv2d-813 [-1, 336, 14, 14] 112,896

SeparableConv2d-814 [-1, 336, 14, 14] 0

BatchNorm2d-815 [-1, 336, 14, 14] 672

BranchSeparables-816 [-1, 336, 14, 14] 0

NormalCell-817 [-1, 2016, 14, 14] 0

ReLU-818 [-1, 2016, 14, 14] 0

Conv2d-819 [-1, 336, 14, 14] 677,376

BatchNorm2d-820 [-1, 336, 14, 14] 672

ActConvBn-821 [-1, 336, 14, 14] 0

ReLU-822 [-1, 2016, 14, 14] 0

Conv2d-823 [-1, 336, 14, 14] 677,376

BatchNorm2d-824 [-1, 336, 14, 14] 672

ActConvBn-825 [-1, 336, 14, 14] 0

ReLU-826 [-1, 336, 14, 14] 0

Conv2d-827 [-1, 336, 14, 14] 8,400

Conv2d-828 [-1, 336, 14, 14] 112,896

SeparableConv2d-829 [-1, 336, 14, 14] 0

BatchNorm2d-830 [-1, 336, 14, 14] 672

ReLU-831 [-1, 336, 14, 14] 0

Conv2d-832 [-1, 336, 14, 14] 8,400

Conv2d-833 [-1, 336, 14, 14] 112,896

SeparableConv2d-834 [-1, 336, 14, 14] 0

BatchNorm2d-835 [-1, 336, 14, 14] 672

BranchSeparables-836 [-1, 336, 14, 14] 0

ReLU-837 [-1, 336, 14, 14] 0

Conv2d-838 [-1, 336, 14, 14] 3,024

Conv2d-839 [-1, 336, 14, 14] 112,896

SeparableConv2d-840 [-1, 336, 14, 14] 0

BatchNorm2d-841 [-1, 336, 14, 14] 672

ReLU-842 [-1, 336, 14, 14] 0

Conv2d-843 [-1, 336, 14, 14] 3,024

Conv2d-844 [-1, 336, 14, 14] 112,896

SeparableConv2d-845 [-1, 336, 14, 14] 0

BatchNorm2d-846 [-1, 336, 14, 14] 672

BranchSeparables-847 [-1, 336, 14, 14] 0

ReLU-848 [-1, 336, 14, 14] 0

Conv2d-849 [-1, 336, 14, 14] 8,400

Conv2d-850 [-1, 336, 14, 14] 112,896

SeparableConv2d-851 [-1, 336, 14, 14] 0

BatchNorm2d-852 [-1, 336, 14, 14] 672

ReLU-853 [-1, 336, 14, 14] 0

Conv2d-854 [-1, 336, 14, 14] 8,400

Conv2d-855 [-1, 336, 14, 14] 112,896

SeparableConv2d-856 [-1, 336, 14, 14] 0

BatchNorm2d-857 [-1, 336, 14, 14] 672

BranchSeparables-858 [-1, 336, 14, 14] 0

ReLU-859 [-1, 336, 14, 14] 0

Conv2d-860 [-1, 336, 14, 14] 3,024

Conv2d-861 [-1, 336, 14, 14] 112,896

SeparableConv2d-862 [-1, 336, 14, 14] 0

BatchNorm2d-863 [-1, 336, 14, 14] 672

ReLU-864 [-1, 336, 14, 14] 0

Conv2d-865 [-1, 336, 14, 14] 3,024

Conv2d-866 [-1, 336, 14, 14] 112,896

SeparableConv2d-867 [-1, 336, 14, 14] 0

BatchNorm2d-868 [-1, 336, 14, 14] 672

BranchSeparables-869 [-1, 336, 14, 14] 0

AvgPool2d-870 [-1, 336, 14, 14] 0

AvgPool2d-871 [-1, 336, 14, 14] 0

AvgPool2d-872 [-1, 336, 14, 14] 0

ReLU-873 [-1, 336, 14, 14] 0

Conv2d-874 [-1, 336, 14, 14] 3,024

Conv2d-875 [-1, 336, 14, 14] 112,896

SeparableConv2d-876 [-1, 336, 14, 14] 0

BatchNorm2d-877 [-1, 336, 14, 14] 672

ReLU-878 [-1, 336, 14, 14] 0

Conv2d-879 [-1, 336, 14, 14] 3,024

Conv2d-880 [-1, 336, 14, 14] 112,896

SeparableConv2d-881 [-1, 336, 14, 14] 0

BatchNorm2d-882 [-1, 336, 14, 14] 672

BranchSeparables-883 [-1, 336, 14, 14] 0

NormalCell-884 [-1, 2016, 14, 14] 0

ReLU-885 [-1, 2016, 14, 14] 0

Conv2d-886 [-1, 336, 14, 14] 677,376

BatchNorm2d-887 [-1, 336, 14, 14] 672

ActConvBn-888 [-1, 336, 14, 14] 0

ReLU-889 [-1, 2016, 14, 14] 0

Conv2d-890 [-1, 336, 14, 14] 677,376

BatchNorm2d-891 [-1, 336, 14, 14] 672

ActConvBn-892 [-1, 336, 14, 14] 0

ReLU-893 [-1, 336, 14, 14] 0

Conv2d-894 [-1, 336, 14, 14] 8,400

Conv2d-895 [-1, 336, 14, 14] 112,896

SeparableConv2d-896 [-1, 336, 14, 14] 0

BatchNorm2d-897 [-1, 336, 14, 14] 672

ReLU-898 [-1, 336, 14, 14] 0

Conv2d-899 [-1, 336, 14, 14] 8,400

Conv2d-900 [-1, 336, 14, 14] 112,896

SeparableConv2d-901 [-1, 336, 14, 14] 0

BatchNorm2d-902 [-1, 336, 14, 14] 672

BranchSeparables-903 [-1, 336, 14, 14] 0

ReLU-904 [-1, 336, 14, 14] 0

Conv2d-905 [-1, 336, 14, 14] 3,024

Conv2d-906 [-1, 336, 14, 14] 112,896

SeparableConv2d-907 [-1, 336, 14, 14] 0

BatchNorm2d-908 [-1, 336, 14, 14] 672

ReLU-909 [-1, 336, 14, 14] 0

Conv2d-910 [-1, 336, 14, 14] 3,024

Conv2d-911 [-1, 336, 14, 14] 112,896

SeparableConv2d-912 [-1, 336, 14, 14] 0

BatchNorm2d-913 [-1, 336, 14, 14] 672

BranchSeparables-914 [-1, 336, 14, 14] 0

ReLU-915 [-1, 336, 14, 14] 0

Conv2d-916 [-1, 336, 14, 14] 8,400

Conv2d-917 [-1, 336, 14, 14] 112,896

SeparableConv2d-918 [-1, 336, 14, 14] 0

BatchNorm2d-919 [-1, 336, 14, 14] 672

ReLU-920 [-1, 336, 14, 14] 0

Conv2d-921 [-1, 336, 14, 14] 8,400

Conv2d-922 [-1, 336, 14, 14] 112,896

SeparableConv2d-923 [-1, 336, 14, 14] 0

BatchNorm2d-924 [-1, 336, 14, 14] 672

BranchSeparables-925 [-1, 336, 14, 14] 0

ReLU-926 [-1, 336, 14, 14] 0

Conv2d-927 [-1, 336, 14, 14] 3,024

Conv2d-928 [-1, 336, 14, 14] 112,896

SeparableConv2d-929 [-1, 336, 14, 14] 0

BatchNorm2d-930 [-1, 336, 14, 14] 672

ReLU-931 [-1, 336, 14, 14] 0

Conv2d-932 [-1, 336, 14, 14] 3,024

Conv2d-933 [-1, 336, 14, 14] 112,896

SeparableConv2d-934 [-1, 336, 14, 14] 0

BatchNorm2d-935 [-1, 336, 14, 14] 672

BranchSeparables-936 [-1, 336, 14, 14] 0

AvgPool2d-937 [-1, 336, 14, 14] 0

AvgPool2d-938 [-1, 336, 14, 14] 0

AvgPool2d-939 [-1, 336, 14, 14] 0

ReLU-940 [-1, 336, 14, 14] 0

Conv2d-941 [-1, 336, 14, 14] 3,024

Conv2d-942 [-1, 336, 14, 14] 112,896

SeparableConv2d-943 [-1, 336, 14, 14] 0

BatchNorm2d-944 [-1, 336, 14, 14] 672

ReLU-945 [-1, 336, 14, 14] 0

Conv2d-946 [-1, 336, 14, 14] 3,024

Conv2d-947 [-1, 336, 14, 14] 112,896

SeparableConv2d-948 [-1, 336, 14, 14] 0

BatchNorm2d-949 [-1, 336, 14, 14] 672

BranchSeparables-950 [-1, 336, 14, 14] 0

NormalCell-951 [-1, 2016, 14, 14] 0

ReLU-952 [-1, 2016, 14, 14] 0

Conv2d-953 [-1, 336, 14, 14] 677,376

BatchNorm2d-954 [-1, 336, 14, 14] 672

ActConvBn-955 [-1, 336, 14, 14] 0

ReLU-956 [-1, 2016, 14, 14] 0

Conv2d-957 [-1, 336, 14, 14] 677,376

BatchNorm2d-958 [-1, 336, 14, 14] 672

ActConvBn-959 [-1, 336, 14, 14] 0

ReLU-960 [-1, 336, 14, 14] 0

Conv2d-961 [-1, 336, 14, 14] 8,400

Conv2d-962 [-1, 336, 14, 14] 112,896

SeparableConv2d-963 [-1, 336, 14, 14] 0

BatchNorm2d-964 [-1, 336, 14, 14] 672

ReLU-965 [-1, 336, 14, 14] 0

Conv2d-966 [-1, 336, 14, 14] 8,400

Conv2d-967 [-1, 336, 14, 14] 112,896

SeparableConv2d-968 [-1, 336, 14, 14] 0

BatchNorm2d-969 [-1, 336, 14, 14] 672

BranchSeparables-970 [-1, 336, 14, 14] 0

ReLU-971 [-1, 336, 14, 14] 0

Conv2d-972 [-1, 336, 14, 14] 3,024

Conv2d-973 [-1, 336, 14, 14] 112,896

SeparableConv2d-974 [-1, 336, 14, 14] 0

BatchNorm2d-975 [-1, 336, 14, 14] 672

ReLU-976 [-1, 336, 14, 14] 0

Conv2d-977 [-1, 336, 14, 14] 3,024

Conv2d-978 [-1, 336, 14, 14] 112,896

SeparableConv2d-979 [-1, 336, 14, 14] 0

BatchNorm2d-980 [-1, 336, 14, 14] 672

BranchSeparables-981 [-1, 336, 14, 14] 0

ReLU-982 [-1, 336, 14, 14] 0

Conv2d-983 [-1, 336, 14, 14] 8,400

Conv2d-984 [-1, 336, 14, 14] 112,896

SeparableConv2d-985 [-1, 336, 14, 14] 0

BatchNorm2d-986 [-1, 336, 14, 14] 672

ReLU-987 [-1, 336, 14, 14] 0

Conv2d-988 [-1, 336, 14, 14] 8,400

Conv2d-989 [-1, 336, 14, 14] 112,896

SeparableConv2d-990 [-1, 336, 14, 14] 0

BatchNorm2d-991 [-1, 336, 14, 14] 672

BranchSeparables-992 [-1, 336, 14, 14] 0

ReLU-993 [-1, 336, 14, 14] 0

Conv2d-994 [-1, 336, 14, 14] 3,024

Conv2d-995 [-1, 336, 14, 14] 112,896

SeparableConv2d-996 [-1, 336, 14, 14] 0

BatchNorm2d-997 [-1, 336, 14, 14] 672

ReLU-998 [-1, 336, 14, 14] 0

Conv2d-999 [-1, 336, 14, 14] 3,024

Conv2d-1000 [-1, 336, 14, 14] 112,896

SeparableConv2d-1001 [-1, 336, 14, 14] 0

BatchNorm2d-1002 [-1, 336, 14, 14] 672

BranchSeparables-1003 [-1, 336, 14, 14] 0

AvgPool2d-1004 [-1, 336, 14, 14] 0

AvgPool2d-1005 [-1, 336, 14, 14] 0

AvgPool2d-1006 [-1, 336, 14, 14] 0

ReLU-1007 [-1, 336, 14, 14] 0

Conv2d-1008 [-1, 336, 14, 14] 3,024

Conv2d-1009 [-1, 336, 14, 14] 112,896

SeparableConv2d-1010 [-1, 336, 14, 14] 0

BatchNorm2d-1011 [-1, 336, 14, 14] 672

ReLU-1012 [-1, 336, 14, 14] 0

Conv2d-1013 [-1, 336, 14, 14] 3,024

Conv2d-1014 [-1, 336, 14, 14] 112,896

SeparableConv2d-1015 [-1, 336, 14, 14] 0

BatchNorm2d-1016 [-1, 336, 14, 14] 672

BranchSeparables-1017 [-1, 336, 14, 14] 0

NormalCell-1018 [-1, 2016, 14, 14] 0

ReLU-1019 [-1, 2016, 14, 14] 0

Conv2d-1020 [-1, 672, 14, 14] 1,354,752

BatchNorm2d-1021 [-1, 672, 14, 14] 1,344

ActConvBn-1022 [-1, 672, 14, 14] 0

ReLU-1023 [-1, 2016, 14, 14] 0

Conv2d-1024 [-1, 672, 14, 14] 1,354,752

BatchNorm2d-1025 [-1, 672, 14, 14] 1,344

ActConvBn-1026 [-1, 672, 14, 14] 0

ReLU-1027 [-1, 672, 14, 14] 0

Conv2dSame-1028 [-1, 672, 7, 7] 16,800

Conv2d-1029 [-1, 672, 7, 7] 451,584

SeparableConv2d-1030 [-1, 672, 7, 7] 0

BatchNorm2d-1031 [-1, 672, 7, 7] 1,344

ReLU-1032 [-1, 672, 7, 7] 0

Conv2d-1033 [-1, 672, 7, 7] 16,800

Conv2d-1034 [-1, 672, 7, 7] 451,584

SeparableConv2d-1035 [-1, 672, 7, 7] 0

BatchNorm2d-1036 [-1, 672, 7, 7] 1,344

BranchSeparables-1037 [-1, 672, 7, 7] 0

ReLU-1038 [-1, 672, 14, 14] 0

Conv2dSame-1039 [-1, 672, 7, 7] 32,928

Conv2d-1040 [-1, 672, 7, 7] 451,584

SeparableConv2d-1041 [-1, 672, 7, 7] 0

BatchNorm2d-1042 [-1, 672, 7, 7] 1,344

ReLU-1043 [-1, 672, 7, 7] 0

Conv2d-1044 [-1, 672, 7, 7] 32,928

Conv2d-1045 [-1, 672, 7, 7] 451,584

SeparableConv2d-1046 [-1, 672, 7, 7] 0

BatchNorm2d-1047 [-1, 672, 7, 7] 1,344

BranchSeparables-1048 [-1, 672, 7, 7] 0

MaxPool2dSame-1049 [-1, 672, 7, 7] 0

ReLU-1050 [-1, 672, 14, 14] 0

Conv2dSame-1051 [-1, 672, 7, 7] 32,928

Conv2d-1052 [-1, 672, 7, 7] 451,584

SeparableConv2d-1053 [-1, 672, 7, 7] 0

BatchNorm2d-1054 [-1, 672, 7, 7] 1,344

ReLU-1055 [-1, 672, 7, 7] 0

Conv2d-1056 [-1, 672, 7, 7] 32,928

Conv2d-1057 [-1, 672, 7, 7] 451,584

SeparableConv2d-1058 [-1, 672, 7, 7] 0

BatchNorm2d-1059 [-1, 672, 7, 7] 1,344

BranchSeparables-1060 [-1, 672, 7, 7] 0

AvgPool2dSame-1061 [-1, 672, 7, 7] 0

ReLU-1062 [-1, 672, 14, 14] 0

Conv2dSame-1063 [-1, 672, 7, 7] 16,800

Conv2d-1064 [-1, 672, 7, 7] 451,584

SeparableConv2d-1065 [-1, 672, 7, 7] 0

BatchNorm2d-1066 [-1, 672, 7, 7] 1,344

ReLU-1067 [-1, 672, 7, 7] 0

Conv2d-1068 [-1, 672, 7, 7] 16,800

Conv2d-1069 [-1, 672, 7, 7] 451,584

SeparableConv2d-1070 [-1, 672, 7, 7] 0

BatchNorm2d-1071 [-1, 672, 7, 7] 1,344

BranchSeparables-1072 [-1, 672, 7, 7] 0

AvgPool2d-1073 [-1, 672, 7, 7] 0

ReLU-1074 [-1, 672, 7, 7] 0

Conv2d-1075 [-1, 672, 7, 7] 6,048

Conv2d-1076 [-1, 672, 7, 7] 451,584

SeparableConv2d-1077 [-1, 672, 7, 7] 0

BatchNorm2d-1078 [-1, 672, 7, 7] 1,344

ReLU-1079 [-1, 672, 7, 7] 0

Conv2d-1080 [-1, 672, 7, 7] 6,048

Conv2d-1081 [-1, 672, 7, 7] 451,584

SeparableConv2d-1082 [-1, 672, 7, 7] 0

BatchNorm2d-1083 [-1, 672, 7, 7] 1,344

BranchSeparables-1084 [-1, 672, 7, 7] 0

MaxPool2dSame-1085 [-1, 672, 7, 7] 0

ReductionCell1-1086 [-1, 2688, 7, 7] 0

ReLU-1087 [-1, 2016, 14, 14] 0

AvgPool2d-1088 [-1, 2016, 7, 7] 0

Conv2d-1089 [-1, 336, 7, 7] 677,376

ZeroPad2d-1090 [-1, 2016, 14, 14] 0

AvgPool2d-1091 [-1, 2016, 7, 7] 0

Conv2d-1092 [-1, 336, 7, 7] 677,376

BatchNorm2d-1093 [-1, 672, 7, 7] 1,344

ReLU-1094 [-1, 2688, 7, 7] 0

Conv2d-1095 [-1, 672, 7, 7] 1,806,336

BatchNorm2d-1096 [-1, 672, 7, 7] 1,344

ActConvBn-1097 [-1, 672, 7, 7] 0

ReLU-1098 [-1, 672, 7, 7] 0

Conv2d-1099 [-1, 672, 7, 7] 16,800

Conv2d-1100 [-1, 672, 7, 7] 451,584

SeparableConv2d-1101 [-1, 672, 7, 7] 0

BatchNorm2d-1102 [-1, 672, 7, 7] 1,344

ReLU-1103 [-1, 672, 7, 7] 0

Conv2d-1104 [-1, 672, 7, 7] 16,800

Conv2d-1105 [-1, 672, 7, 7] 451,584

SeparableConv2d-1106 [-1, 672, 7, 7] 0

BatchNorm2d-1107 [-1, 672, 7, 7] 1,344

BranchSeparables-1108 [-1, 672, 7, 7] 0

ReLU-1109 [-1, 672, 7, 7] 0

Conv2d-1110 [-1, 672, 7, 7] 6,048

Conv2d-1111 [-1, 672, 7, 7] 451,584

SeparableConv2d-1112 [-1, 672, 7, 7] 0

BatchNorm2d-1113 [-1, 672, 7, 7] 1,344

ReLU-1114 [-1, 672, 7, 7] 0

Conv2d-1115 [-1, 672, 7, 7] 6,048

Conv2d-1116 [-1, 672, 7, 7] 451,584

SeparableConv2d-1117 [-1, 672, 7, 7] 0

BatchNorm2d-1118 [-1, 672, 7, 7] 1,344

BranchSeparables-1119 [-1, 672, 7, 7] 0

ReLU-1120 [-1, 672, 7, 7] 0

Conv2d-1121 [-1, 672, 7, 7] 16,800

Conv2d-1122 [-1, 672, 7, 7] 451,584

SeparableConv2d-1123 [-1, 672, 7, 7] 0

BatchNorm2d-1124 [-1, 672, 7, 7] 1,344

ReLU-1125 [-1, 672, 7, 7] 0

Conv2d-1126 [-1, 672, 7, 7] 16,800

Conv2d-1127 [-1, 672, 7, 7] 451,584

SeparableConv2d-1128 [-1, 672, 7, 7] 0

BatchNorm2d-1129 [-1, 672, 7, 7] 1,344

BranchSeparables-1130 [-1, 672, 7, 7] 0

ReLU-1131 [-1, 672, 7, 7] 0

Conv2d-1132 [-1, 672, 7, 7] 6,048

Conv2d-1133 [-1, 672, 7, 7] 451,584

SeparableConv2d-1134 [-1, 672, 7, 7] 0

BatchNorm2d-1135 [-1, 672, 7, 7] 1,344

ReLU-1136 [-1, 672, 7, 7] 0

Conv2d-1137 [-1, 672, 7, 7] 6,048

Conv2d-1138 [-1, 672, 7, 7] 451,584

SeparableConv2d-1139 [-1, 672, 7, 7] 0

BatchNorm2d-1140 [-1, 672, 7, 7] 1,344

BranchSeparables-1141 [-1, 672, 7, 7] 0

AvgPool2d-1142 [-1, 672, 7, 7] 0

AvgPool2d-1143 [-1, 672, 7, 7] 0

AvgPool2d-1144 [-1, 672, 7, 7] 0

ReLU-1145 [-1, 672, 7, 7] 0

Conv2d-1146 [-1, 672, 7, 7] 6,048

Conv2d-1147 [-1, 672, 7, 7] 451,584

SeparableConv2d-1148 [-1, 672, 7, 7] 0

BatchNorm2d-1149 [-1, 672, 7, 7] 1,344

ReLU-1150 [-1, 672, 7, 7] 0

Conv2d-1151 [-1, 672, 7, 7] 6,048

Conv2d-1152 [-1, 672, 7, 7] 451,584

SeparableConv2d-1153 [-1, 672, 7, 7] 0

BatchNorm2d-1154 [-1, 672, 7, 7] 1,344

BranchSeparables-1155 [-1, 672, 7, 7] 0

FirstCell-1156 [-1, 4032, 7, 7] 0

ReLU-1157 [-1, 2688, 7, 7] 0

Conv2d-1158 [-1, 672, 7, 7] 1,806,336

BatchNorm2d-1159 [-1, 672, 7, 7] 1,344

ActConvBn-1160 [-1, 672, 7, 7] 0

ReLU-1161 [-1, 4032, 7, 7] 0

Conv2d-1162 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1163 [-1, 672, 7, 7] 1,344

ActConvBn-1164 [-1, 672, 7, 7] 0

ReLU-1165 [-1, 672, 7, 7] 0

Conv2d-1166 [-1, 672, 7, 7] 16,800

Conv2d-1167 [-1, 672, 7, 7] 451,584

SeparableConv2d-1168 [-1, 672, 7, 7] 0

BatchNorm2d-1169 [-1, 672, 7, 7] 1,344

ReLU-1170 [-1, 672, 7, 7] 0

Conv2d-1171 [-1, 672, 7, 7] 16,800

Conv2d-1172 [-1, 672, 7, 7] 451,584

SeparableConv2d-1173 [-1, 672, 7, 7] 0

BatchNorm2d-1174 [-1, 672, 7, 7] 1,344

BranchSeparables-1175 [-1, 672, 7, 7] 0

ReLU-1176 [-1, 672, 7, 7] 0

Conv2d-1177 [-1, 672, 7, 7] 6,048

Conv2d-1178 [-1, 672, 7, 7] 451,584

SeparableConv2d-1179 [-1, 672, 7, 7] 0

BatchNorm2d-1180 [-1, 672, 7, 7] 1,344

ReLU-1181 [-1, 672, 7, 7] 0

Conv2d-1182 [-1, 672, 7, 7] 6,048

Conv2d-1183 [-1, 672, 7, 7] 451,584

SeparableConv2d-1184 [-1, 672, 7, 7] 0

BatchNorm2d-1185 [-1, 672, 7, 7] 1,344

BranchSeparables-1186 [-1, 672, 7, 7] 0

ReLU-1187 [-1, 672, 7, 7] 0

Conv2d-1188 [-1, 672, 7, 7] 16,800

Conv2d-1189 [-1, 672, 7, 7] 451,584

SeparableConv2d-1190 [-1, 672, 7, 7] 0

BatchNorm2d-1191 [-1, 672, 7, 7] 1,344

ReLU-1192 [-1, 672, 7, 7] 0

Conv2d-1193 [-1, 672, 7, 7] 16,800

Conv2d-1194 [-1, 672, 7, 7] 451,584

SeparableConv2d-1195 [-1, 672, 7, 7] 0

BatchNorm2d-1196 [-1, 672, 7, 7] 1,344

BranchSeparables-1197 [-1, 672, 7, 7] 0

ReLU-1198 [-1, 672, 7, 7] 0

Conv2d-1199 [-1, 672, 7, 7] 6,048

Conv2d-1200 [-1, 672, 7, 7] 451,584

SeparableConv2d-1201 [-1, 672, 7, 7] 0

BatchNorm2d-1202 [-1, 672, 7, 7] 1,344

ReLU-1203 [-1, 672, 7, 7] 0

Conv2d-1204 [-1, 672, 7, 7] 6,048

Conv2d-1205 [-1, 672, 7, 7] 451,584

SeparableConv2d-1206 [-1, 672, 7, 7] 0

BatchNorm2d-1207 [-1, 672, 7, 7] 1,344

BranchSeparables-1208 [-1, 672, 7, 7] 0

AvgPool2d-1209 [-1, 672, 7, 7] 0

AvgPool2d-1210 [-1, 672, 7, 7] 0

AvgPool2d-1211 [-1, 672, 7, 7] 0

ReLU-1212 [-1, 672, 7, 7] 0

Conv2d-1213 [-1, 672, 7, 7] 6,048

Conv2d-1214 [-1, 672, 7, 7] 451,584

SeparableConv2d-1215 [-1, 672, 7, 7] 0

BatchNorm2d-1216 [-1, 672, 7, 7] 1,344

ReLU-1217 [-1, 672, 7, 7] 0

Conv2d-1218 [-1, 672, 7, 7] 6,048

Conv2d-1219 [-1, 672, 7, 7] 451,584

SeparableConv2d-1220 [-1, 672, 7, 7] 0

BatchNorm2d-1221 [-1, 672, 7, 7] 1,344

BranchSeparables-1222 [-1, 672, 7, 7] 0

NormalCell-1223 [-1, 4032, 7, 7] 0

ReLU-1224 [-1, 4032, 7, 7] 0

Conv2d-1225 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1226 [-1, 672, 7, 7] 1,344

ActConvBn-1227 [-1, 672, 7, 7] 0

ReLU-1228 [-1, 4032, 7, 7] 0

Conv2d-1229 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1230 [-1, 672, 7, 7] 1,344

ActConvBn-1231 [-1, 672, 7, 7] 0

ReLU-1232 [-1, 672, 7, 7] 0

Conv2d-1233 [-1, 672, 7, 7] 16,800

Conv2d-1234 [-1, 672, 7, 7] 451,584

SeparableConv2d-1235 [-1, 672, 7, 7] 0

BatchNorm2d-1236 [-1, 672, 7, 7] 1,344

ReLU-1237 [-1, 672, 7, 7] 0

Conv2d-1238 [-1, 672, 7, 7] 16,800

Conv2d-1239 [-1, 672, 7, 7] 451,584

SeparableConv2d-1240 [-1, 672, 7, 7] 0

BatchNorm2d-1241 [-1, 672, 7, 7] 1,344

BranchSeparables-1242 [-1, 672, 7, 7] 0

ReLU-1243 [-1, 672, 7, 7] 0

Conv2d-1244 [-1, 672, 7, 7] 6,048

Conv2d-1245 [-1, 672, 7, 7] 451,584

SeparableConv2d-1246 [-1, 672, 7, 7] 0

BatchNorm2d-1247 [-1, 672, 7, 7] 1,344

ReLU-1248 [-1, 672, 7, 7] 0

Conv2d-1249 [-1, 672, 7, 7] 6,048

Conv2d-1250 [-1, 672, 7, 7] 451,584

SeparableConv2d-1251 [-1, 672, 7, 7] 0

BatchNorm2d-1252 [-1, 672, 7, 7] 1,344

BranchSeparables-1253 [-1, 672, 7, 7] 0

ReLU-1254 [-1, 672, 7, 7] 0

Conv2d-1255 [-1, 672, 7, 7] 16,800

Conv2d-1256 [-1, 672, 7, 7] 451,584

SeparableConv2d-1257 [-1, 672, 7, 7] 0

BatchNorm2d-1258 [-1, 672, 7, 7] 1,344

ReLU-1259 [-1, 672, 7, 7] 0

Conv2d-1260 [-1, 672, 7, 7] 16,800

Conv2d-1261 [-1, 672, 7, 7] 451,584

SeparableConv2d-1262 [-1, 672, 7, 7] 0

BatchNorm2d-1263 [-1, 672, 7, 7] 1,344

BranchSeparables-1264 [-1, 672, 7, 7] 0

ReLU-1265 [-1, 672, 7, 7] 0

Conv2d-1266 [-1, 672, 7, 7] 6,048

Conv2d-1267 [-1, 672, 7, 7] 451,584

SeparableConv2d-1268 [-1, 672, 7, 7] 0

BatchNorm2d-1269 [-1, 672, 7, 7] 1,344

ReLU-1270 [-1, 672, 7, 7] 0

Conv2d-1271 [-1, 672, 7, 7] 6,048

Conv2d-1272 [-1, 672, 7, 7] 451,584

SeparableConv2d-1273 [-1, 672, 7, 7] 0

BatchNorm2d-1274 [-1, 672, 7, 7] 1,344

BranchSeparables-1275 [-1, 672, 7, 7] 0

AvgPool2d-1276 [-1, 672, 7, 7] 0

AvgPool2d-1277 [-1, 672, 7, 7] 0

AvgPool2d-1278 [-1, 672, 7, 7] 0

ReLU-1279 [-1, 672, 7, 7] 0

Conv2d-1280 [-1, 672, 7, 7] 6,048

Conv2d-1281 [-1, 672, 7, 7] 451,584

SeparableConv2d-1282 [-1, 672, 7, 7] 0

BatchNorm2d-1283 [-1, 672, 7, 7] 1,344

ReLU-1284 [-1, 672, 7, 7] 0

Conv2d-1285 [-1, 672, 7, 7] 6,048

Conv2d-1286 [-1, 672, 7, 7] 451,584

SeparableConv2d-1287 [-1, 672, 7, 7] 0

BatchNorm2d-1288 [-1, 672, 7, 7] 1,344

BranchSeparables-1289 [-1, 672, 7, 7] 0

NormalCell-1290 [-1, 4032, 7, 7] 0

ReLU-1291 [-1, 4032, 7, 7] 0

Conv2d-1292 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1293 [-1, 672, 7, 7] 1,344

ActConvBn-1294 [-1, 672, 7, 7] 0

ReLU-1295 [-1, 4032, 7, 7] 0

Conv2d-1296 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1297 [-1, 672, 7, 7] 1,344

ActConvBn-1298 [-1, 672, 7, 7] 0

ReLU-1299 [-1, 672, 7, 7] 0

Conv2d-1300 [-1, 672, 7, 7] 16,800

Conv2d-1301 [-1, 672, 7, 7] 451,584

SeparableConv2d-1302 [-1, 672, 7, 7] 0

BatchNorm2d-1303 [-1, 672, 7, 7] 1,344

ReLU-1304 [-1, 672, 7, 7] 0

Conv2d-1305 [-1, 672, 7, 7] 16,800

Conv2d-1306 [-1, 672, 7, 7] 451,584

SeparableConv2d-1307 [-1, 672, 7, 7] 0

BatchNorm2d-1308 [-1, 672, 7, 7] 1,344

BranchSeparables-1309 [-1, 672, 7, 7] 0

ReLU-1310 [-1, 672, 7, 7] 0

Conv2d-1311 [-1, 672, 7, 7] 6,048

Conv2d-1312 [-1, 672, 7, 7] 451,584

SeparableConv2d-1313 [-1, 672, 7, 7] 0

BatchNorm2d-1314 [-1, 672, 7, 7] 1,344

ReLU-1315 [-1, 672, 7, 7] 0

Conv2d-1316 [-1, 672, 7, 7] 6,048

Conv2d-1317 [-1, 672, 7, 7] 451,584

SeparableConv2d-1318 [-1, 672, 7, 7] 0

BatchNorm2d-1319 [-1, 672, 7, 7] 1,344

BranchSeparables-1320 [-1, 672, 7, 7] 0

ReLU-1321 [-1, 672, 7, 7] 0

Conv2d-1322 [-1, 672, 7, 7] 16,800

Conv2d-1323 [-1, 672, 7, 7] 451,584

SeparableConv2d-1324 [-1, 672, 7, 7] 0

BatchNorm2d-1325 [-1, 672, 7, 7] 1,344

ReLU-1326 [-1, 672, 7, 7] 0

Conv2d-1327 [-1, 672, 7, 7] 16,800

Conv2d-1328 [-1, 672, 7, 7] 451,584

SeparableConv2d-1329 [-1, 672, 7, 7] 0

BatchNorm2d-1330 [-1, 672, 7, 7] 1,344

BranchSeparables-1331 [-1, 672, 7, 7] 0

ReLU-1332 [-1, 672, 7, 7] 0

Conv2d-1333 [-1, 672, 7, 7] 6,048

Conv2d-1334 [-1, 672, 7, 7] 451,584

SeparableConv2d-1335 [-1, 672, 7, 7] 0

BatchNorm2d-1336 [-1, 672, 7, 7] 1,344

ReLU-1337 [-1, 672, 7, 7] 0

Conv2d-1338 [-1, 672, 7, 7] 6,048

Conv2d-1339 [-1, 672, 7, 7] 451,584

SeparableConv2d-1340 [-1, 672, 7, 7] 0

BatchNorm2d-1341 [-1, 672, 7, 7] 1,344

BranchSeparables-1342 [-1, 672, 7, 7] 0

AvgPool2d-1343 [-1, 672, 7, 7] 0

AvgPool2d-1344 [-1, 672, 7, 7] 0

AvgPool2d-1345 [-1, 672, 7, 7] 0

ReLU-1346 [-1, 672, 7, 7] 0

Conv2d-1347 [-1, 672, 7, 7] 6,048

Conv2d-1348 [-1, 672, 7, 7] 451,584

SeparableConv2d-1349 [-1, 672, 7, 7] 0

BatchNorm2d-1350 [-1, 672, 7, 7] 1,344

ReLU-1351 [-1, 672, 7, 7] 0

Conv2d-1352 [-1, 672, 7, 7] 6,048

Conv2d-1353 [-1, 672, 7, 7] 451,584

SeparableConv2d-1354 [-1, 672, 7, 7] 0

BatchNorm2d-1355 [-1, 672, 7, 7] 1,344

BranchSeparables-1356 [-1, 672, 7, 7] 0

NormalCell-1357 [-1, 4032, 7, 7] 0

ReLU-1358 [-1, 4032, 7, 7] 0

Conv2d-1359 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1360 [-1, 672, 7, 7] 1,344

ActConvBn-1361 [-1, 672, 7, 7] 0

ReLU-1362 [-1, 4032, 7, 7] 0

Conv2d-1363 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1364 [-1, 672, 7, 7] 1,344

ActConvBn-1365 [-1, 672, 7, 7] 0

ReLU-1366 [-1, 672, 7, 7] 0

Conv2d-1367 [-1, 672, 7, 7] 16,800

Conv2d-1368 [-1, 672, 7, 7] 451,584

SeparableConv2d-1369 [-1, 672, 7, 7] 0

BatchNorm2d-1370 [-1, 672, 7, 7] 1,344

ReLU-1371 [-1, 672, 7, 7] 0

Conv2d-1372 [-1, 672, 7, 7] 16,800

Conv2d-1373 [-1, 672, 7, 7] 451,584

SeparableConv2d-1374 [-1, 672, 7, 7] 0

BatchNorm2d-1375 [-1, 672, 7, 7] 1,344

BranchSeparables-1376 [-1, 672, 7, 7] 0

ReLU-1377 [-1, 672, 7, 7] 0

Conv2d-1378 [-1, 672, 7, 7] 6,048

Conv2d-1379 [-1, 672, 7, 7] 451,584

SeparableConv2d-1380 [-1, 672, 7, 7] 0

BatchNorm2d-1381 [-1, 672, 7, 7] 1,344

ReLU-1382 [-1, 672, 7, 7] 0

Conv2d-1383 [-1, 672, 7, 7] 6,048

Conv2d-1384 [-1, 672, 7, 7] 451,584

SeparableConv2d-1385 [-1, 672, 7, 7] 0

BatchNorm2d-1386 [-1, 672, 7, 7] 1,344

BranchSeparables-1387 [-1, 672, 7, 7] 0

ReLU-1388 [-1, 672, 7, 7] 0

Conv2d-1389 [-1, 672, 7, 7] 16,800

Conv2d-1390 [-1, 672, 7, 7] 451,584

SeparableConv2d-1391 [-1, 672, 7, 7] 0

BatchNorm2d-1392 [-1, 672, 7, 7] 1,344

ReLU-1393 [-1, 672, 7, 7] 0

Conv2d-1394 [-1, 672, 7, 7] 16,800

Conv2d-1395 [-1, 672, 7, 7] 451,584

SeparableConv2d-1396 [-1, 672, 7, 7] 0

BatchNorm2d-1397 [-1, 672, 7, 7] 1,344

BranchSeparables-1398 [-1, 672, 7, 7] 0

ReLU-1399 [-1, 672, 7, 7] 0

Conv2d-1400 [-1, 672, 7, 7] 6,048

Conv2d-1401 [-1, 672, 7, 7] 451,584

SeparableConv2d-1402 [-1, 672, 7, 7] 0

BatchNorm2d-1403 [-1, 672, 7, 7] 1,344

ReLU-1404 [-1, 672, 7, 7] 0

Conv2d-1405 [-1, 672, 7, 7] 6,048

Conv2d-1406 [-1, 672, 7, 7] 451,584

SeparableConv2d-1407 [-1, 672, 7, 7] 0

BatchNorm2d-1408 [-1, 672, 7, 7] 1,344

BranchSeparables-1409 [-1, 672, 7, 7] 0

AvgPool2d-1410 [-1, 672, 7, 7] 0

AvgPool2d-1411 [-1, 672, 7, 7] 0

AvgPool2d-1412 [-1, 672, 7, 7] 0

ReLU-1413 [-1, 672, 7, 7] 0

Conv2d-1414 [-1, 672, 7, 7] 6,048

Conv2d-1415 [-1, 672, 7, 7] 451,584

SeparableConv2d-1416 [-1, 672, 7, 7] 0

BatchNorm2d-1417 [-1, 672, 7, 7] 1,344

ReLU-1418 [-1, 672, 7, 7] 0

Conv2d-1419 [-1, 672, 7, 7] 6,048

Conv2d-1420 [-1, 672, 7, 7] 451,584

SeparableConv2d-1421 [-1, 672, 7, 7] 0

BatchNorm2d-1422 [-1, 672, 7, 7] 1,344

BranchSeparables-1423 [-1, 672, 7, 7] 0

NormalCell-1424 [-1, 4032, 7, 7] 0

ReLU-1425 [-1, 4032, 7, 7] 0

Conv2d-1426 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1427 [-1, 672, 7, 7] 1,344

ActConvBn-1428 [-1, 672, 7, 7] 0

ReLU-1429 [-1, 4032, 7, 7] 0

Conv2d-1430 [-1, 672, 7, 7] 2,709,504

BatchNorm2d-1431 [-1, 672, 7, 7] 1,344

ActConvBn-1432 [-1, 672, 7, 7] 0

ReLU-1433 [-1, 672, 7, 7] 0

Conv2d-1434 [-1, 672, 7, 7] 16,800

Conv2d-1435 [-1, 672, 7, 7] 451,584

SeparableConv2d-1436 [-1, 672, 7, 7] 0

BatchNorm2d-1437 [-1, 672, 7, 7] 1,344

ReLU-1438 [-1, 672, 7, 7] 0

Conv2d-1439 [-1, 672, 7, 7] 16,800

Conv2d-1440 [-1, 672, 7, 7] 451,584

SeparableConv2d-1441 [-1, 672, 7, 7] 0

BatchNorm2d-1442 [-1, 672, 7, 7] 1,344

BranchSeparables-1443 [-1, 672, 7, 7] 0

ReLU-1444 [-1, 672, 7, 7] 0

Conv2d-1445 [-1, 672, 7, 7] 6,048

Conv2d-1446 [-1, 672, 7, 7] 451,584

SeparableConv2d-1447 [-1, 672, 7, 7] 0

BatchNorm2d-1448 [-1, 672, 7, 7] 1,344

ReLU-1449 [-1, 672, 7, 7] 0

Conv2d-1450 [-1, 672, 7, 7] 6,048

Conv2d-1451 [-1, 672, 7, 7] 451,584

SeparableConv2d-1452 [-1, 672, 7, 7] 0

BatchNorm2d-1453 [-1, 672, 7, 7] 1,344

BranchSeparables-1454 [-1, 672, 7, 7] 0

ReLU-1455 [-1, 672, 7, 7] 0

Conv2d-1456 [-1, 672, 7, 7] 16,800

Conv2d-1457 [-1, 672, 7, 7] 451,584

SeparableConv2d-1458 [-1, 672, 7, 7] 0

BatchNorm2d-1459 [-1, 672, 7, 7] 1,344

ReLU-1460 [-1, 672, 7, 7] 0

Conv2d-1461 [-1, 672, 7, 7] 16,800

Conv2d-1462 [-1, 672, 7, 7] 451,584

SeparableConv2d-1463 [-1, 672, 7, 7] 0

BatchNorm2d-1464 [-1, 672, 7, 7] 1,344

BranchSeparables-1465 [-1, 672, 7, 7] 0

ReLU-1466 [-1, 672, 7, 7] 0

Conv2d-1467 [-1, 672, 7, 7] 6,048

Conv2d-1468 [-1, 672, 7, 7] 451,584

SeparableConv2d-1469 [-1, 672, 7, 7] 0

BatchNorm2d-1470 [-1, 672, 7, 7] 1,344

ReLU-1471 [-1, 672, 7, 7] 0

Conv2d-1472 [-1, 672, 7, 7] 6,048

Conv2d-1473 [-1, 672, 7, 7] 451,584

SeparableConv2d-1474 [-1, 672, 7, 7] 0

BatchNorm2d-1475 [-1, 672, 7, 7] 1,344

BranchSeparables-1476 [-1, 672, 7, 7] 0

AvgPool2d-1477 [-1, 672, 7, 7] 0

AvgPool2d-1478 [-1, 672, 7, 7] 0

AvgPool2d-1479 [-1, 672, 7, 7] 0

ReLU-1480 [-1, 672, 7, 7] 0

Conv2d-1481 [-1, 672, 7, 7] 6,048

Conv2d-1482 [-1, 672, 7, 7] 451,584

SeparableConv2d-1483 [-1, 672, 7, 7] 0

BatchNorm2d-1484 [-1, 672, 7, 7] 1,344

ReLU-1485 [-1, 672, 7, 7] 0

Conv2d-1486 [-1, 672, 7, 7] 6,048

Conv2d-1487 [-1, 672, 7, 7] 451,584

SeparableConv2d-1488 [-1, 672, 7, 7] 0

BatchNorm2d-1489 [-1, 672, 7, 7] 1,344

BranchSeparables-1490 [-1, 672, 7, 7] 0

NormalCell-1491 [-1, 4032, 7, 7] 0

ReLU-1492 [-1, 4032, 7, 7] 0

AdaptiveAvgPool2d-1493 [-1, 4032, 1, 1] 0

Flatten-1494 [-1, 4032] 0

SelectAdaptivePool2d-1495 [-1, 4032] 0

Dropout-1496 [-1, 4032] 0

Linear-1497 [-1, 101] 407,333

================================================================

Total params: 85,127,483

Trainable params: 407,333

Non-trainable params: 84,720,150

----------------------------------------------------------------

Input size (MB): 0.57

Forward/backward pass size (MB): 1203.40

Params size (MB): 324.74

Estimated Total Size (MB): 1528.71

1. Fine-tuning Process for MobileNetV3

features.0.0.weight: requires\_grad=False

features.0.1.weight: requires\_grad=False

features.0.1.bias: requires\_grad=False

features.1.block.0.0.weight: requires\_grad=False

features.1.block.0.1.weight: requires\_grad=False

features.1.block.0.1.bias: requires\_grad=False

features.1.block.1.0.weight: requires\_grad=False

features.1.block.1.1.weight: requires\_grad=False

features.1.block.1.1.bias: requires\_grad=False

features.2.block.0.0.weight: requires\_grad=False

features.2.block.0.1.weight: requires\_grad=False

features.2.block.0.1.bias: requires\_grad=False

features.2.block.1.0.weight: requires\_grad=False

features.2.block.1.1.weight: requires\_grad=False

features.2.block.1.1.bias: requires\_grad=False

features.2.block.2.0.weight: requires\_grad=False

features.2.block.2.1.weight: requires\_grad=False

features.2.block.2.1.bias: requires\_grad=False

features.3.block.0.0.weight: requires\_grad=False

features.3.block.0.1.weight: requires\_grad=False

features.3.block.0.1.bias: requires\_grad=False

features.3.block.1.0.weight: requires\_grad=False

features.3.block.1.1.weight: requires\_grad=False

features.3.block.1.1.bias: requires\_grad=False

features.3.block.2.0.weight: requires\_grad=False

features.3.block.2.1.weight: requires\_grad=False

features.3.block.2.1.bias: requires\_grad=False

features.4.block.0.0.weight: requires\_grad=False

features.4.block.0.1.weight: requires\_grad=False

features.4.block.0.1.bias: requires\_grad=False

features.4.block.1.0.weight: requires\_grad=False

features.4.block.1.1.weight: requires\_grad=False

features.4.block.1.1.bias: requires\_grad=False

features.4.block.2.fc1.weight: requires\_grad=False

features.4.block.2.fc1.bias: requires\_grad=False

features.4.block.2.fc2.weight: requires\_grad=False

features.4.block.2.fc2.bias: requires\_grad=False

features.4.block.3.0.weight: requires\_grad=False

features.4.block.3.1.weight: requires\_grad=False

features.4.block.3.1.bias: requires\_grad=False

features.5.block.0.0.weight: requires\_grad=False

features.5.block.0.1.weight: requires\_grad=False

features.5.block.0.1.bias: requires\_grad=False

features.5.block.1.0.weight: requires\_grad=False

features.5.block.1.1.weight: requires\_grad=False

features.5.block.1.1.bias: requires\_grad=False

features.5.block.2.fc1.weight: requires\_grad=False

features.5.block.2.fc1.bias: requires\_grad=False

features.5.block.2.fc2.weight: requires\_grad=False

features.5.block.2.fc2.bias: requires\_grad=False

features.5.block.3.0.weight: requires\_grad=False

features.5.block.3.1.weight: requires\_grad=False

features.5.block.3.1.bias: requires\_grad=False

features.6.block.0.0.weight: requires\_grad=False

features.6.block.0.1.weight: requires\_grad=False

features.6.block.0.1.bias: requires\_grad=False

features.6.block.1.0.weight: requires\_grad=False

features.6.block.1.1.weight: requires\_grad=False

features.6.block.1.1.bias: requires\_grad=False

features.6.block.2.fc1.weight: requires\_grad=False

features.6.block.2.fc1.bias: requires\_grad=False

features.6.block.2.fc2.weight: requires\_grad=False

features.6.block.2.fc2.bias: requires\_grad=False

features.6.block.3.0.weight: requires\_grad=False

features.6.block.3.1.weight: requires\_grad=False

features.6.block.3.1.bias: requires\_grad=False

features.7.block.0.0.weight: requires\_grad=False

features.7.block.0.1.weight: requires\_grad=False

features.7.block.0.1.bias: requires\_grad=False

features.7.block.1.0.weight: requires\_grad=False

features.7.block.1.1.weight: requires\_grad=False

features.7.block.1.1.bias: requires\_grad=False

features.7.block.2.0.weight: requires\_grad=False

features.7.block.2.1.weight: requires\_grad=False

features.7.block.2.1.bias: requires\_grad=False

features.8.block.0.0.weight: requires\_grad=False

features.8.block.0.1.weight: requires\_grad=False

features.8.block.0.1.bias: requires\_grad=False

features.8.block.1.0.weight: requires\_grad=False

features.8.block.1.1.weight: requires\_grad=False

features.8.block.1.1.bias: requires\_grad=False

features.8.block.2.0.weight: requires\_grad=False

features.8.block.2.1.weight: requires\_grad=False

features.8.block.2.1.bias: requires\_grad=False

features.9.block.0.0.weight: requires\_grad=False

features.9.block.0.1.weight: requires\_grad=False

features.9.block.0.1.bias: requires\_grad=False

features.9.block.1.0.weight: requires\_grad=False

features.9.block.1.1.weight: requires\_grad=False

features.9.block.1.1.bias: requires\_grad=False

features.9.block.2.0.weight: requires\_grad=False

features.9.block.2.1.weight: requires\_grad=False

features.9.block.2.1.bias: requires\_grad=False

features.10.block.0.0.weight: requires\_grad=False

features.10.block.0.1.weight: requires\_grad=False

features.10.block.0.1.bias: requires\_grad=False

features.10.block.1.0.weight: requires\_grad=False

features.10.block.1.1.weight: requires\_grad=False

features.10.block.1.1.bias: requires\_grad=False

features.10.block.2.0.weight: requires\_grad=False

features.10.block.2.1.weight: requires\_grad=False

features.10.block.2.1.bias: requires\_grad=False

features.11.block.0.0.weight: requires\_grad=False

features.11.block.0.1.weight: requires\_grad=False

features.11.block.0.1.bias: requires\_grad=False

features.11.block.1.0.weight: requires\_grad=False

features.11.block.1.1.weight: requires\_grad=False

features.11.block.1.1.bias: requires\_grad=False

features.11.block.2.fc1.weight: requires\_grad=False

features.11.block.2.fc1.bias: requires\_grad=False

features.11.block.2.fc2.weight: requires\_grad=False

features.11.block.2.fc2.bias: requires\_grad=False

features.11.block.3.0.weight: requires\_grad=False

features.11.block.3.1.weight: requires\_grad=False

features.11.block.3.1.bias: requires\_grad=True

features.12.block.0.0.weight: requires\_grad=True

features.12.block.0.1.weight: requires\_grad=True

features.12.block.0.1.bias: requires\_grad=True

features.12.block.1.0.weight: requires\_grad=True

features.12.block.1.1.weight: requires\_grad=True

features.12.block.1.1.bias: requires\_grad=True

features.12.block.2.fc1.weight: requires\_grad=True

features.12.block.2.fc1.bias: requires\_grad=True

features.12.block.2.fc2.weight: requires\_grad=True

features.12.block.2.fc2.bias: requires\_grad=True

features.12.block.3.0.weight: requires\_grad=True

features.12.block.3.1.weight: requires\_grad=True

features.12.block.3.1.bias: requires\_grad=True

features.13.block.0.0.weight: requires\_grad=True

features.13.block.0.1.weight: requires\_grad=True

features.13.block.0.1.bias: requires\_grad=True

features.13.block.1.0.weight: requires\_grad=True

features.13.block.1.1.weight: requires\_grad=True

features.13.block.1.1.bias: requires\_grad=True

features.13.block.2.fc1.weight: requires\_grad=True

features.13.block.2.fc1.bias: requires\_grad=True

features.13.block.2.fc2.weight: requires\_grad=True

features.13.block.2.fc2.bias: requires\_grad=True

features.13.block.3.0.weight: requires\_grad=True

features.13.block.3.1.weight: requires\_grad=True

features.13.block.3.1.bias: requires\_grad=True

features.14.block.0.0.weight: requires\_grad=True

features.14.block.0.1.weight: requires\_grad=True

features.14.block.0.1.bias: requires\_grad=True

features.14.block.1.0.weight: requires\_grad=True

features.14.block.1.1.weight: requires\_grad=True

features.14.block.1.1.bias: requires\_grad=True

features.14.block.2.fc1.weight: requires\_grad=True

features.14.block.2.fc1.bias: requires\_grad=True

features.14.block.2.fc2.weight: requires\_grad=True

features.14.block.2.fc2.bias: requires\_grad=True

features.14.block.3.0.weight: requires\_grad=True

features.14.block.3.1.weight: requires\_grad=True

features.14.block.3.1.bias: requires\_grad=True

features.15.block.0.0.weight: requires\_grad=True

features.15.block.0.1.weight: requires\_grad=True

features.15.block.0.1.bias: requires\_grad=True

features.15.block.1.0.weight: requires\_grad=True

features.15.block.1.1.weight: requires\_grad=True

features.15.block.1.1.bias: requires\_grad=True

features.15.block.2.fc1.weight: requires\_grad=True

features.15.block.2.fc1.bias: requires\_grad=True

features.15.block.2.fc2.weight: requires\_grad=True

features.15.block.2.fc2.bias: requires\_grad=True

features.15.block.3.0.weight: requires\_grad=True

features.15.block.3.1.weight: requires\_grad=True

features.15.block.3.1.bias: requires\_grad=True

features.16.0.weight: requires\_grad=True

features.16.1.weight: requires\_grad=True

features.16.1.bias: requires\_grad=True

classifier.0.weight: requires\_grad=True

classifier.0.bias: requires\_grad=True

classifier.3.weight: requires\_grad=True

classifier.3.bias: requires\_grad=True

1. MobileNetV3 Fine tuned model for training summary:

----------------------------------------------------------------

Layer (type) Output Shape Param #

================================================================

Conv2d-1 [-1, 16, 112, 112] 432

BatchNorm2d-2 [-1, 16, 112, 112] 32

Hardswish-3 [-1, 16, 112, 112] 0

Conv2d-4 [-1, 16, 112, 112] 144

BatchNorm2d-5 [-1, 16, 112, 112] 32

ReLU-6 [-1, 16, 112, 112] 0

Conv2d-7 [-1, 16, 112, 112] 256

BatchNorm2d-8 [-1, 16, 112, 112] 32

InvertedResidual-9 [-1, 16, 112, 112] 0

Conv2d-10 [-1, 64, 112, 112] 1,024

BatchNorm2d-11 [-1, 64, 112, 112] 128

ReLU-12 [-1, 64, 112, 112] 0

Conv2d-13 [-1, 64, 56, 56] 576

BatchNorm2d-14 [-1, 64, 56, 56] 128

ReLU-15 [-1, 64, 56, 56] 0

Conv2d-16 [-1, 24, 56, 56] 1,536

BatchNorm2d-17 [-1, 24, 56, 56] 48

InvertedResidual-18 [-1, 24, 56, 56] 0

Conv2d-19 [-1, 72, 56, 56] 1,728

BatchNorm2d-20 [-1, 72, 56, 56] 144

ReLU-21 [-1, 72, 56, 56] 0

Conv2d-22 [-1, 72, 56, 56] 648

BatchNorm2d-23 [-1, 72, 56, 56] 144

ReLU-24 [-1, 72, 56, 56] 0

Conv2d-25 [-1, 24, 56, 56] 1,728

BatchNorm2d-26 [-1, 24, 56, 56] 48

InvertedResidual-27 [-1, 24, 56, 56] 0

Conv2d-28 [-1, 72, 56, 56] 1,728

BatchNorm2d-29 [-1, 72, 56, 56] 144

ReLU-30 [-1, 72, 56, 56] 0

Conv2d-31 [-1, 72, 28, 28] 1,800

BatchNorm2d-32 [-1, 72, 28, 28] 144

ReLU-33 [-1, 72, 28, 28] 0

AdaptiveAvgPool2d-34 [-1, 72, 1, 1] 0

Conv2d-35 [-1, 24, 1, 1] 1,752

ReLU-36 [-1, 24, 1, 1] 0

Conv2d-37 [-1, 72, 1, 1] 1,800

Hardsigmoid-38 [-1, 72, 1, 1] 0

SqueezeExcitation-39 [-1, 72, 28, 28] 0

Conv2d-40 [-1, 40, 28, 28] 2,880

BatchNorm2d-41 [-1, 40, 28, 28] 80

InvertedResidual-42 [-1, 40, 28, 28] 0

Conv2d-43 [-1, 120, 28, 28] 4,800

BatchNorm2d-44 [-1, 120, 28, 28] 240

ReLU-45 [-1, 120, 28, 28] 0

Conv2d-46 [-1, 120, 28, 28] 3,000

BatchNorm2d-47 [-1, 120, 28, 28] 240

ReLU-48 [-1, 120, 28, 28] 0

AdaptiveAvgPool2d-49 [-1, 120, 1, 1] 0

Conv2d-50 [-1, 32, 1, 1] 3,872

ReLU-51 [-1, 32, 1, 1] 0

Conv2d-52 [-1, 120, 1, 1] 3,960

Hardsigmoid-53 [-1, 120, 1, 1] 0

SqueezeExcitation-54 [-1, 120, 28, 28] 0

Conv2d-55 [-1, 40, 28, 28] 4,800

BatchNorm2d-56 [-1, 40, 28, 28] 80

InvertedResidual-57 [-1, 40, 28, 28] 0

Conv2d-58 [-1, 120, 28, 28] 4,800

BatchNorm2d-59 [-1, 120, 28, 28] 240

ReLU-60 [-1, 120, 28, 28] 0

Conv2d-61 [-1, 120, 28, 28] 3,000

BatchNorm2d-62 [-1, 120, 28, 28] 240

ReLU-63 [-1, 120, 28, 28] 0

AdaptiveAvgPool2d-64 [-1, 120, 1, 1] 0

Conv2d-65 [-1, 32, 1, 1] 3,872

ReLU-66 [-1, 32, 1, 1] 0

Conv2d-67 [-1, 120, 1, 1] 3,960

Hardsigmoid-68 [-1, 120, 1, 1] 0

SqueezeExcitation-69 [-1, 120, 28, 28] 0

Conv2d-70 [-1, 40, 28, 28] 4,800

BatchNorm2d-71 [-1, 40, 28, 28] 80

InvertedResidual-72 [-1, 40, 28, 28] 0

Conv2d-73 [-1, 240, 28, 28] 9,600

BatchNorm2d-74 [-1, 240, 28, 28] 480

Hardswish-75 [-1, 240, 28, 28] 0

Conv2d-76 [-1, 240, 14, 14] 2,160

BatchNorm2d-77 [-1, 240, 14, 14] 480

Hardswish-78 [-1, 240, 14, 14] 0

Conv2d-79 [-1, 80, 14, 14] 19,200

BatchNorm2d-80 [-1, 80, 14, 14] 160

InvertedResidual-81 [-1, 80, 14, 14] 0

Conv2d-82 [-1, 200, 14, 14] 16,000

BatchNorm2d-83 [-1, 200, 14, 14] 400

Hardswish-84 [-1, 200, 14, 14] 0

Conv2d-85 [-1, 200, 14, 14] 1,800

BatchNorm2d-86 [-1, 200, 14, 14] 400

Hardswish-87 [-1, 200, 14, 14] 0

Conv2d-88 [-1, 80, 14, 14] 16,000

BatchNorm2d-89 [-1, 80, 14, 14] 160

InvertedResidual-90 [-1, 80, 14, 14] 0

Conv2d-91 [-1, 184, 14, 14] 14,720

BatchNorm2d-92 [-1, 184, 14, 14] 368

Hardswish-93 [-1, 184, 14, 14] 0

Conv2d-94 [-1, 184, 14, 14] 1,656

BatchNorm2d-95 [-1, 184, 14, 14] 368

Hardswish-96 [-1, 184, 14, 14] 0

Conv2d-97 [-1, 80, 14, 14] 14,720

BatchNorm2d-98 [-1, 80, 14, 14] 160

InvertedResidual-99 [-1, 80, 14, 14] 0

Conv2d-100 [-1, 184, 14, 14] 14,720

BatchNorm2d-101 [-1, 184, 14, 14] 368

Hardswish-102 [-1, 184, 14, 14] 0

Conv2d-103 [-1, 184, 14, 14] 1,656

BatchNorm2d-104 [-1, 184, 14, 14] 368

Hardswish-105 [-1, 184, 14, 14] 0

Conv2d-106 [-1, 80, 14, 14] 14,720

BatchNorm2d-107 [-1, 80, 14, 14] 160

InvertedResidual-108 [-1, 80, 14, 14] 0

Conv2d-109 [-1, 480, 14, 14] 38,400

BatchNorm2d-110 [-1, 480, 14, 14] 960

Hardswish-111 [-1, 480, 14, 14] 0

Conv2d-112 [-1, 480, 14, 14] 4,320

BatchNorm2d-113 [-1, 480, 14, 14] 960

Hardswish-114 [-1, 480, 14, 14] 0

AdaptiveAvgPool2d-115 [-1, 480, 1, 1] 0

Conv2d-116 [-1, 120, 1, 1] 57,720

ReLU-117 [-1, 120, 1, 1] 0

Conv2d-118 [-1, 480, 1, 1] 58,080

Hardsigmoid-119 [-1, 480, 1, 1] 0

SqueezeExcitation-120 [-1, 480, 14, 14] 0

Conv2d-121 [-1, 112, 14, 14] 53,760

BatchNorm2d-122 [-1, 112, 14, 14] 224

InvertedResidual-123 [-1, 112, 14, 14] 0

Conv2d-124 [-1, 672, 14, 14] 75,264

BatchNorm2d-125 [-1, 672, 14, 14] 1,344

Hardswish-126 [-1, 672, 14, 14] 0

Conv2d-127 [-1, 672, 14, 14] 6,048

BatchNorm2d-128 [-1, 672, 14, 14] 1,344

Hardswish-129 [-1, 672, 14, 14] 0

AdaptiveAvgPool2d-130 [-1, 672, 1, 1] 0

Conv2d-131 [-1, 168, 1, 1] 113,064

ReLU-132 [-1, 168, 1, 1] 0

Conv2d-133 [-1, 672, 1, 1] 113,568

Hardsigmoid-134 [-1, 672, 1, 1] 0

SqueezeExcitation-135 [-1, 672, 14, 14] 0

Conv2d-136 [-1, 112, 14, 14] 75,264

BatchNorm2d-137 [-1, 112, 14, 14] 224

InvertedResidual-138 [-1, 112, 14, 14] 0

Conv2d-139 [-1, 672, 14, 14] 75,264

BatchNorm2d-140 [-1, 672, 14, 14] 1,344

Hardswish-141 [-1, 672, 14, 14] 0

Conv2d-142 [-1, 672, 7, 7] 16,800

BatchNorm2d-143 [-1, 672, 7, 7] 1,344

Hardswish-144 [-1, 672, 7, 7] 0

AdaptiveAvgPool2d-145 [-1, 672, 1, 1] 0

Conv2d-146 [-1, 168, 1, 1] 113,064

ReLU-147 [-1, 168, 1, 1] 0

Conv2d-148 [-1, 672, 1, 1] 113,568

Hardsigmoid-149 [-1, 672, 1, 1] 0

SqueezeExcitation-150 [-1, 672, 7, 7] 0

Conv2d-151 [-1, 160, 7, 7] 107,520

BatchNorm2d-152 [-1, 160, 7, 7] 320

InvertedResidual-153 [-1, 160, 7, 7] 0

Conv2d-154 [-1, 960, 7, 7] 153,600

BatchNorm2d-155 [-1, 960, 7, 7] 1,920

Hardswish-156 [-1, 960, 7, 7] 0

Conv2d-157 [-1, 960, 7, 7] 24,000

BatchNorm2d-158 [-1, 960, 7, 7] 1,920

Hardswish-159 [-1, 960, 7, 7] 0

AdaptiveAvgPool2d-160 [-1, 960, 1, 1] 0

Conv2d-161 [-1, 240, 1, 1] 230,640

ReLU-162 [-1, 240, 1, 1] 0

Conv2d-163 [-1, 960, 1, 1] 231,360

Hardsigmoid-164 [-1, 960, 1, 1] 0

SqueezeExcitation-165 [-1, 960, 7, 7] 0

Conv2d-166 [-1, 160, 7, 7] 153,600

BatchNorm2d-167 [-1, 160, 7, 7] 320

InvertedResidual-168 [-1, 160, 7, 7] 0

Conv2d-169 [-1, 960, 7, 7] 153,600

BatchNorm2d-170 [-1, 960, 7, 7] 1,920

Hardswish-171 [-1, 960, 7, 7] 0

Conv2d-172 [-1, 960, 7, 7] 24,000

BatchNorm2d-173 [-1, 960, 7, 7] 1,920

Hardswish-174 [-1, 960, 7, 7] 0

AdaptiveAvgPool2d-175 [-1, 960, 1, 1] 0

Conv2d-176 [-1, 240, 1, 1] 230,640

ReLU-177 [-1, 240, 1, 1] 0

Conv2d-178 [-1, 960, 1, 1] 231,360

Hardsigmoid-179 [-1, 960, 1, 1] 0

SqueezeExcitation-180 [-1, 960, 7, 7] 0

Conv2d-181 [-1, 160, 7, 7] 153,600

BatchNorm2d-182 [-1, 160, 7, 7] 320

InvertedResidual-183 [-1, 160, 7, 7] 0

Conv2d-184 [-1, 960, 7, 7] 153,600

BatchNorm2d-185 [-1, 960, 7, 7] 1,920

Hardswish-186 [-1, 960, 7, 7] 0

AdaptiveAvgPool2d-187 [-1, 960, 1, 1] 0

Linear-188 [-1, 1280] 1,230,080

Hardswish-189 [-1, 1280] 0

Dropout-190 [-1, 1280] 0

Linear-191 [-1, 101] 129,381

================================================================

Total params: 4,331,413

Trainable params: 3,925,045

Non-trainable params: 406,368

----------------------------------------------------------------

Input size (MB): 0.57

Forward/backward pass size (MB): 105.41

Params size (MB): 16.52

Estimated Total Size (MB): 122.50

----------------------------------------------------------------

# **REFERENCES**

1. Babu, S., & B. Sai Chandana. (2023). Mobilenetv3: a deep learning technique for human face expressions identification. *International Journal of Information Technology, 15*(6), 3229–3243. <https://doi.org/10.1007/s41870-023-01380-x>
2. Jain, V. (2019, November 22). *Everything you need to know about MobileNetV3 and its comparison with previous versions*. Medium. <https://towardsdatascience.com/everything-you-need-to-know-about-mobilenetv3-and-its-comparison-with-previous-versions-a5d5e5a6eeaa>
3. Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). *A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, 53(8)*. <https://doi.org/10.1007/s10462-020-09825-6>
4. Qian, S., Ning, C., & Hu, Y. (2021). MobileNetV3 for Image Classification. *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE).* <https://doi.org/10.1109/icbaie52039.2021.9389905>
5. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/cvpr.2018.00474>
6. Suharjito, S., Elwirehardja, G. N., & Prayoga, J. S. (2021). Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. *Computers and Electronics in Agriculture, 188*, 106359. https://doi.org/10.1016/j.compag.2021.106359
7. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. *(2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2015.7298594>
8. Zahangir Alom, et al. “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches.” *ArXiv (Cornell University)*, 3 Mar. 2018.
9. Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018, April 11). *Learning transferable architectures for Scalable Image Recognition*. arXiv.org. https://arxiv.org/abs/1707.07012