Module 2: Building Blocks for Image Recognition

Video 8: GoogLeNet / Inception

Analytics Vidhya

GoogLeNet / Inception v1

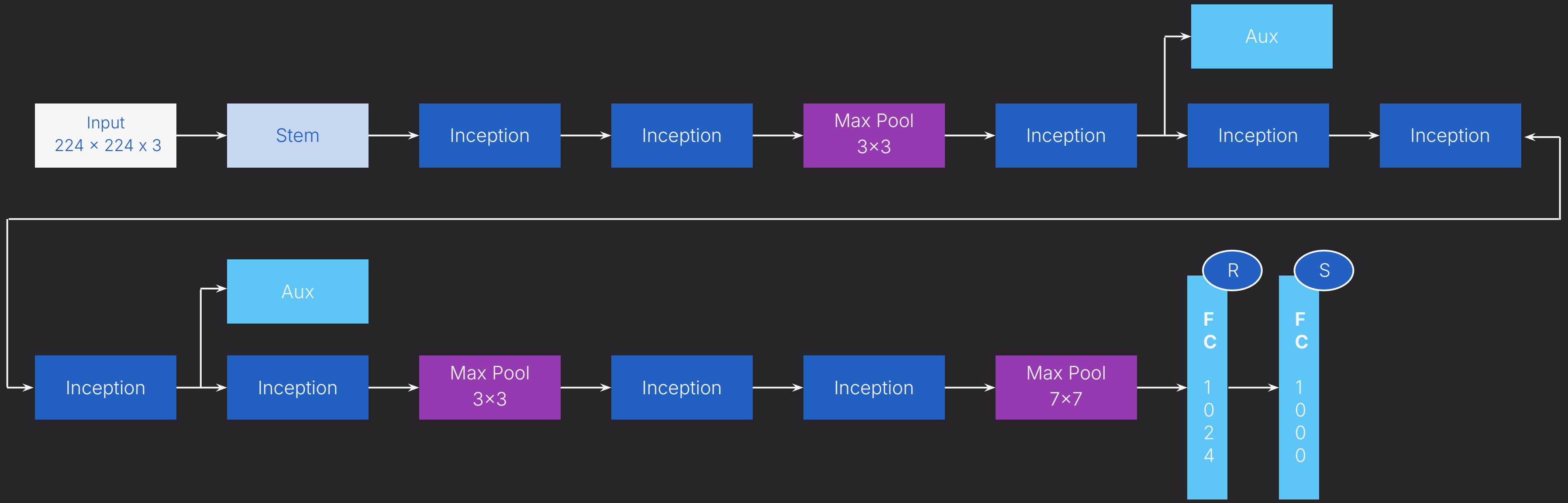
- Developed in the year 2014, in the paper *"Going Deeper with Convolutions"*.

Christian Szegedy

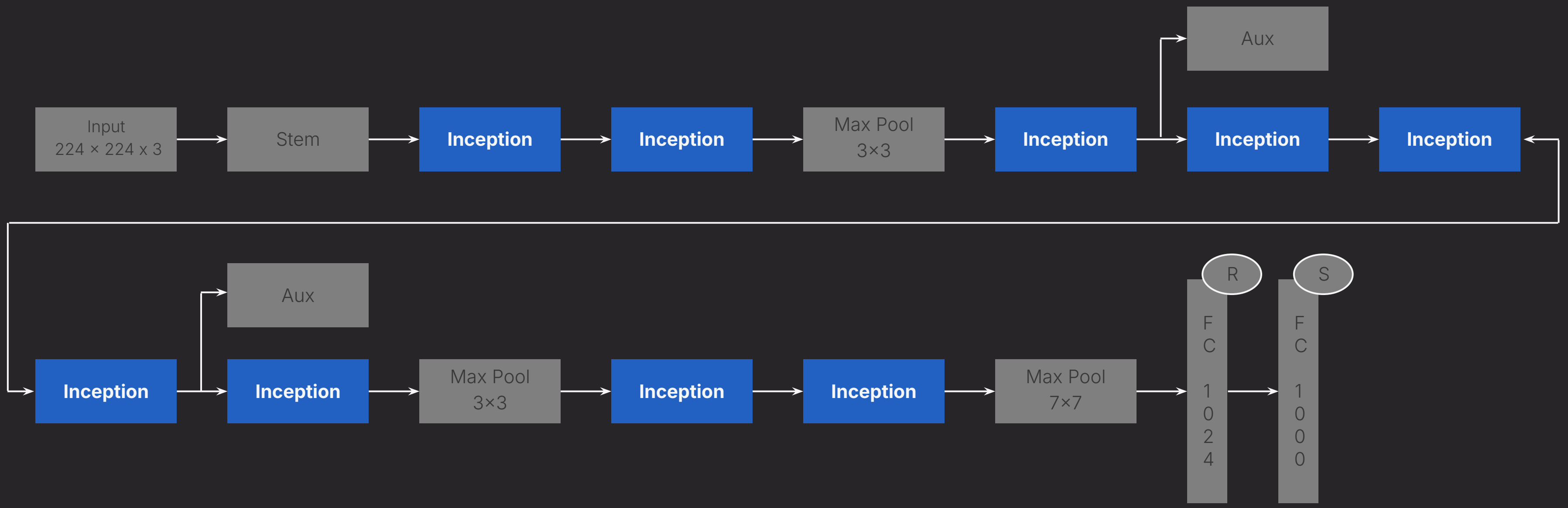
"Going Deeper with Convolutions" - "Inception" Meme

GoogLeNet / Inception v1

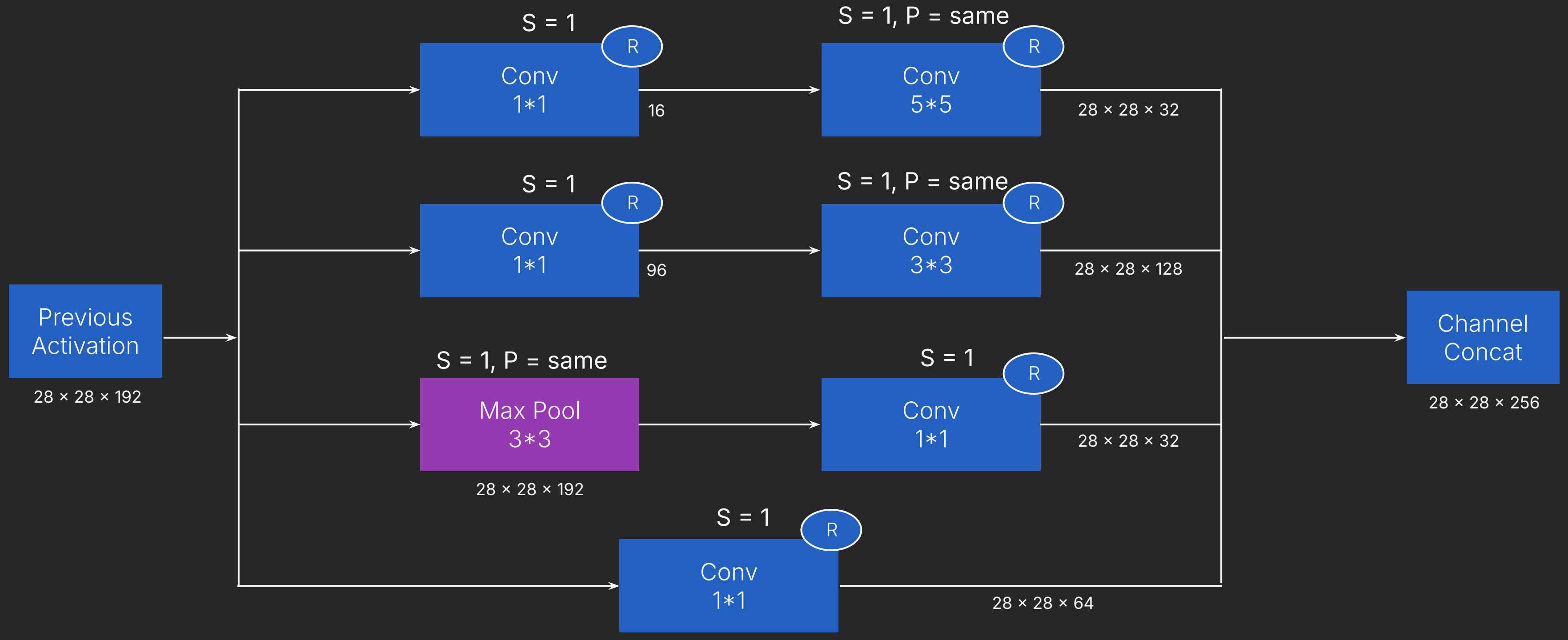
- Introduced the concept of “Network within a network”.



Inception v1: Architecture

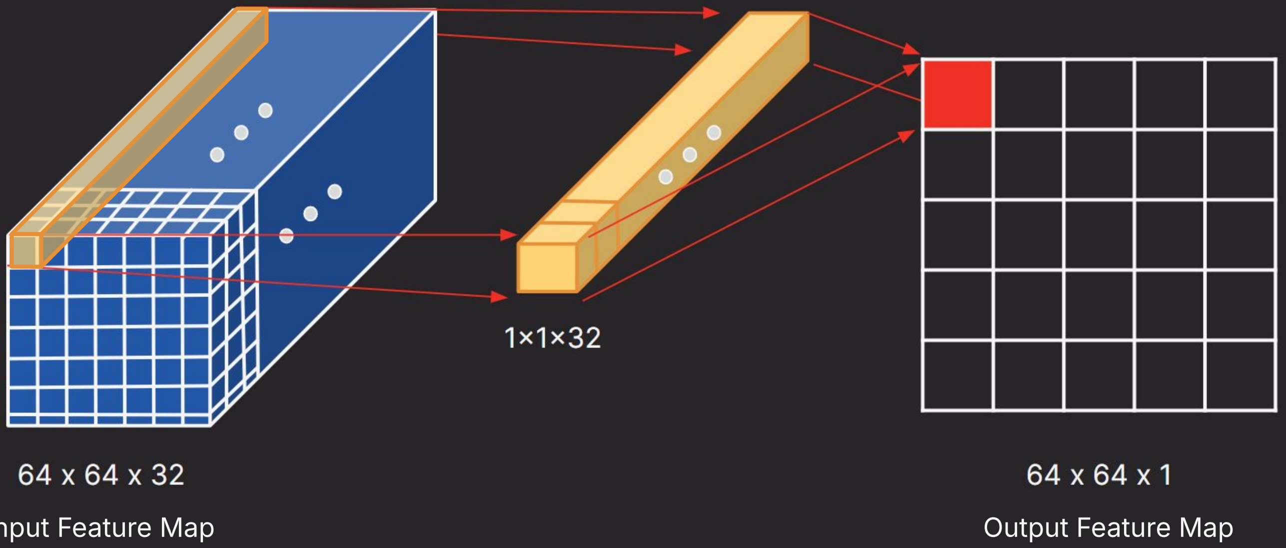


Inception Block



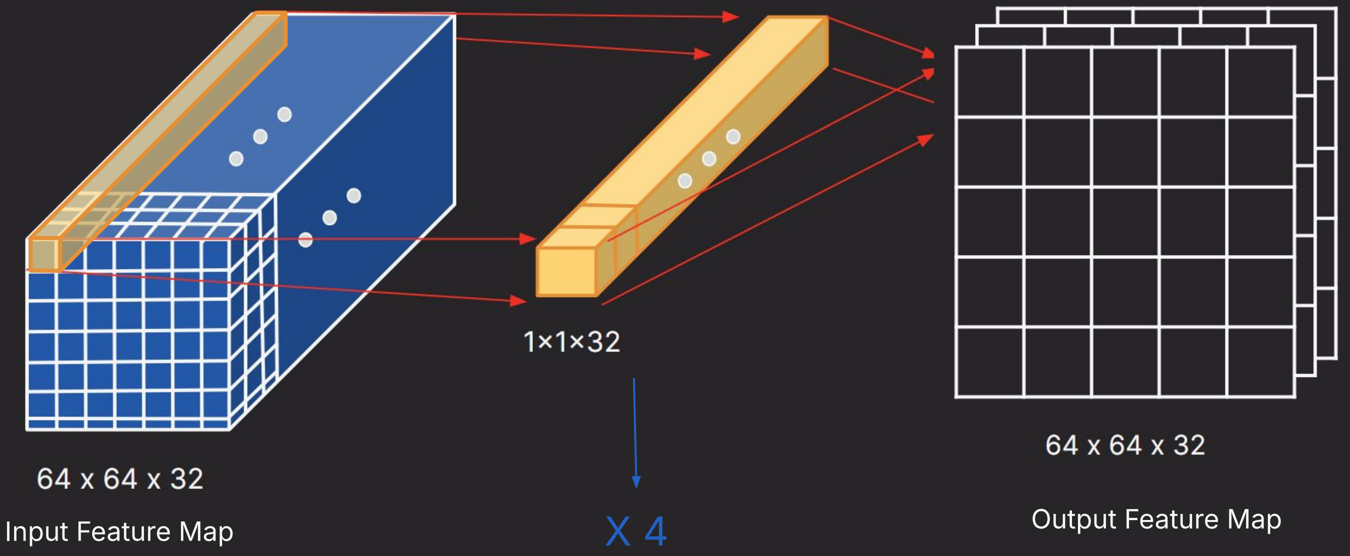
Inception Block

- **1×1 Conv:** Linearly combines multiple input channels into a single output channel.



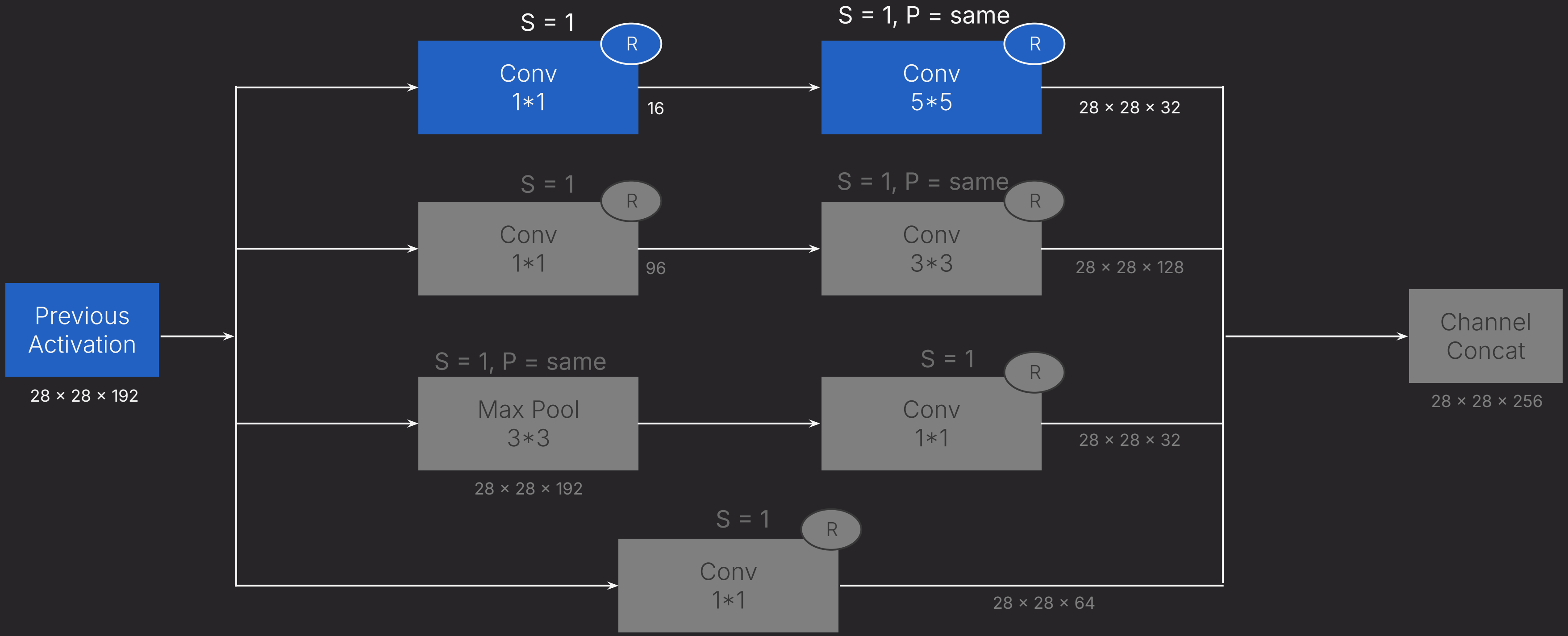
Inception Block

- **1×1 Conv:** Linearly combines multiple input channels into a single output channel.



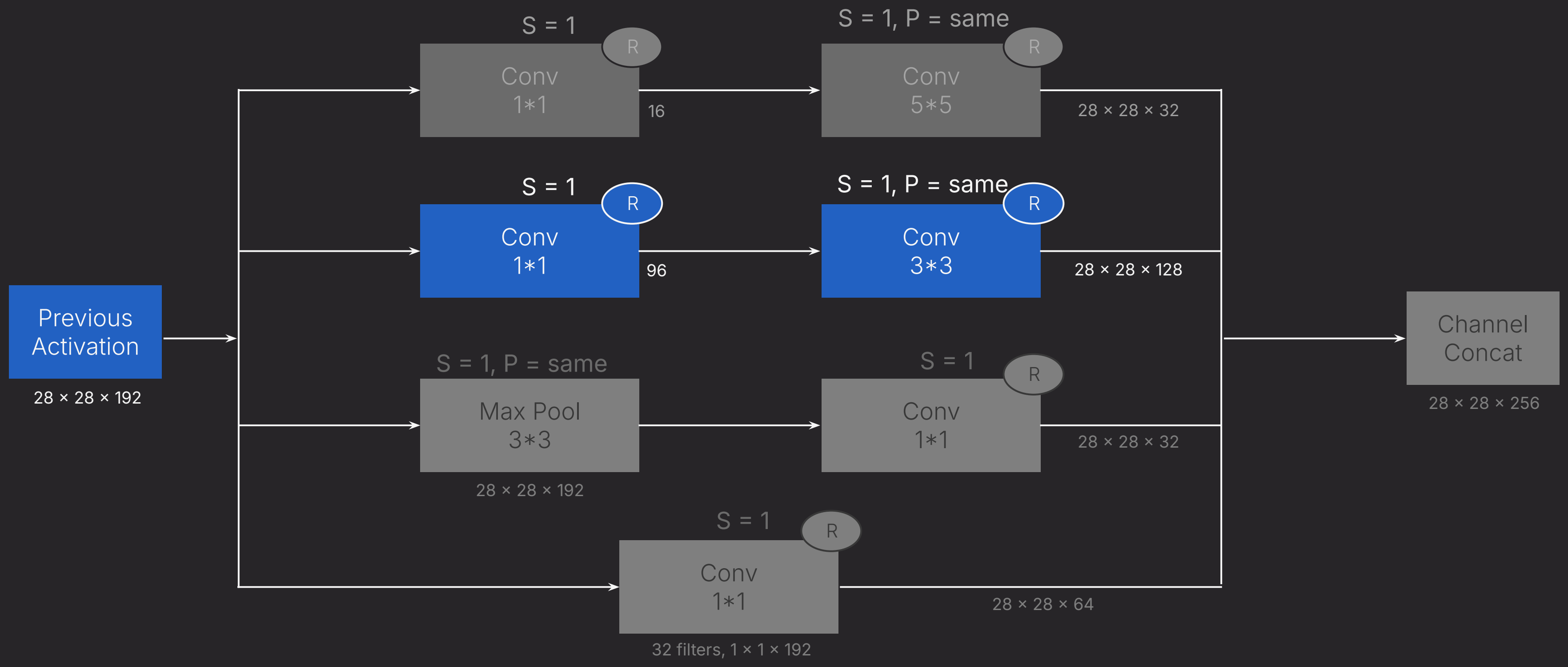
Inception Block

Addition of a 1×1 convolution before a 5×5 convolution reduces the channel depth of the image.



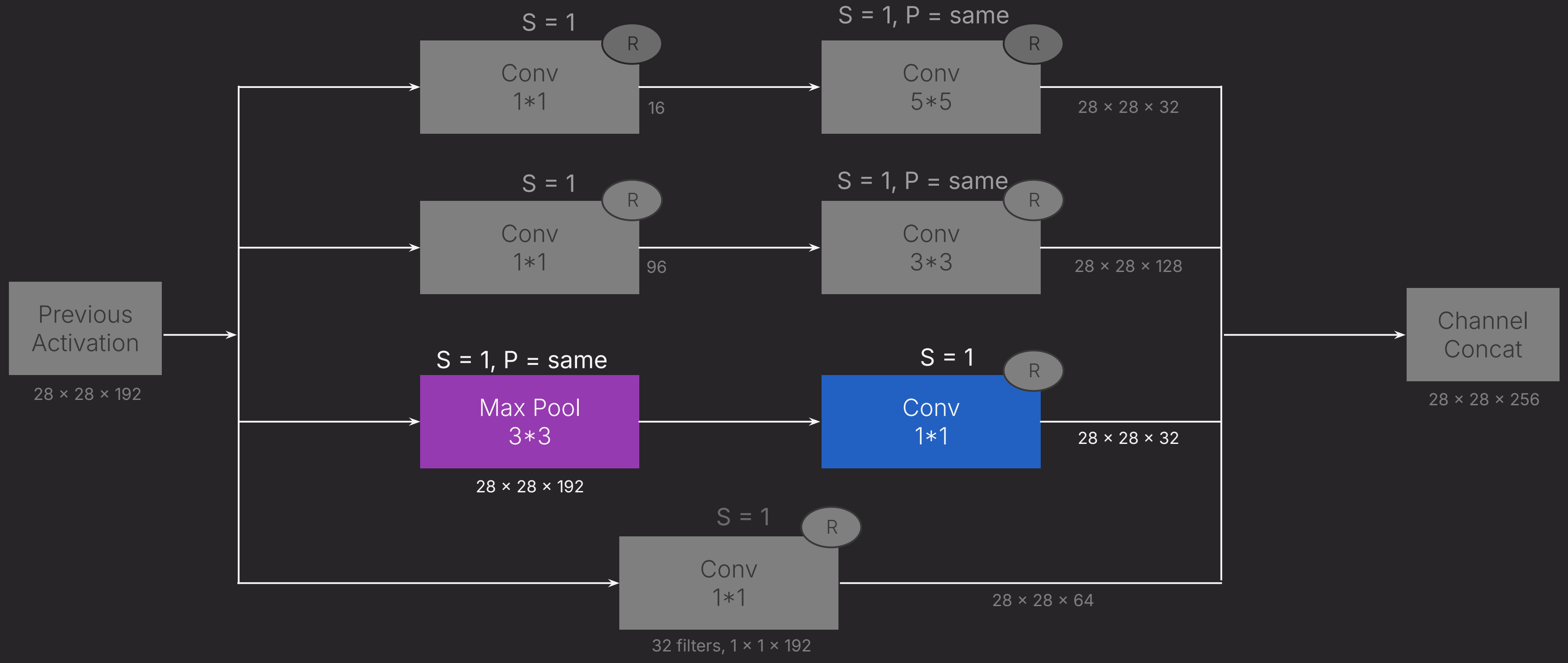
Inception Block

Addition of a 1×1 convolution before a 3×3 convolution reduces the channel depth of the image.



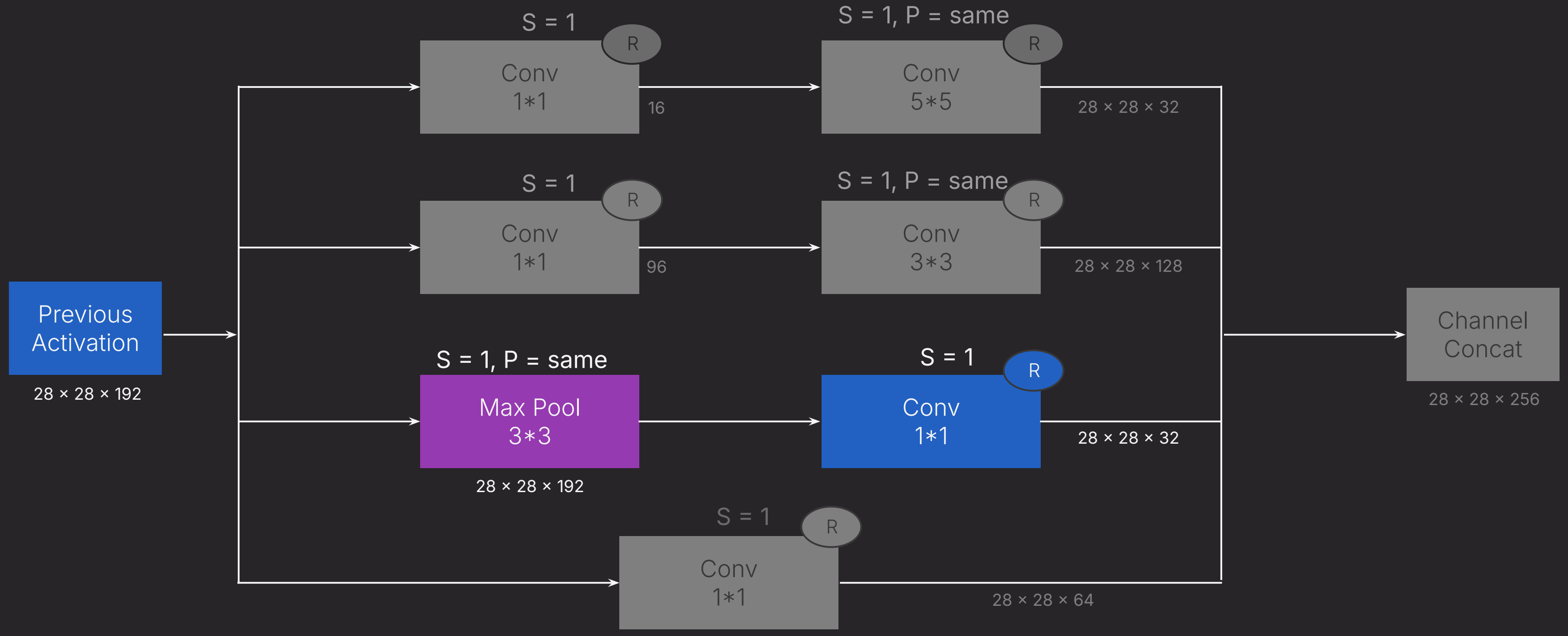
Inception Block

Max Pooling



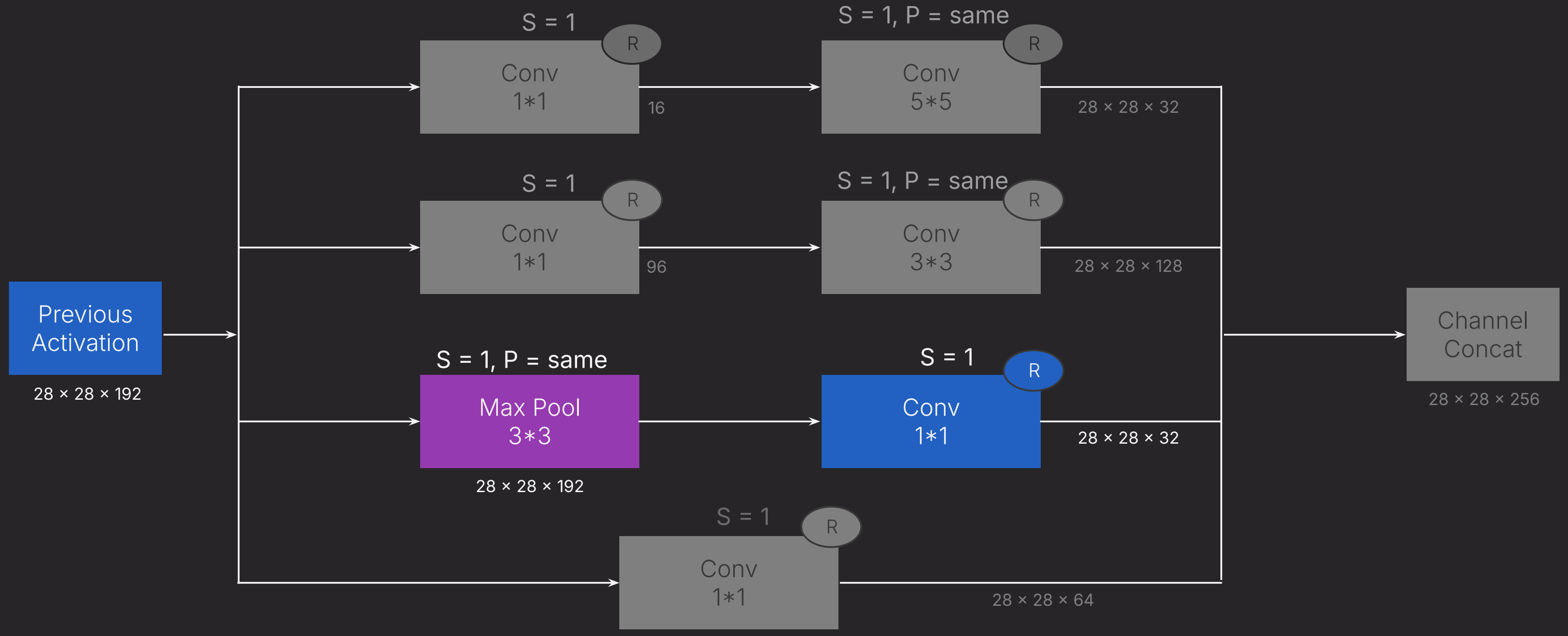
Inception Block

Why is same padding used in Max Pooling ?



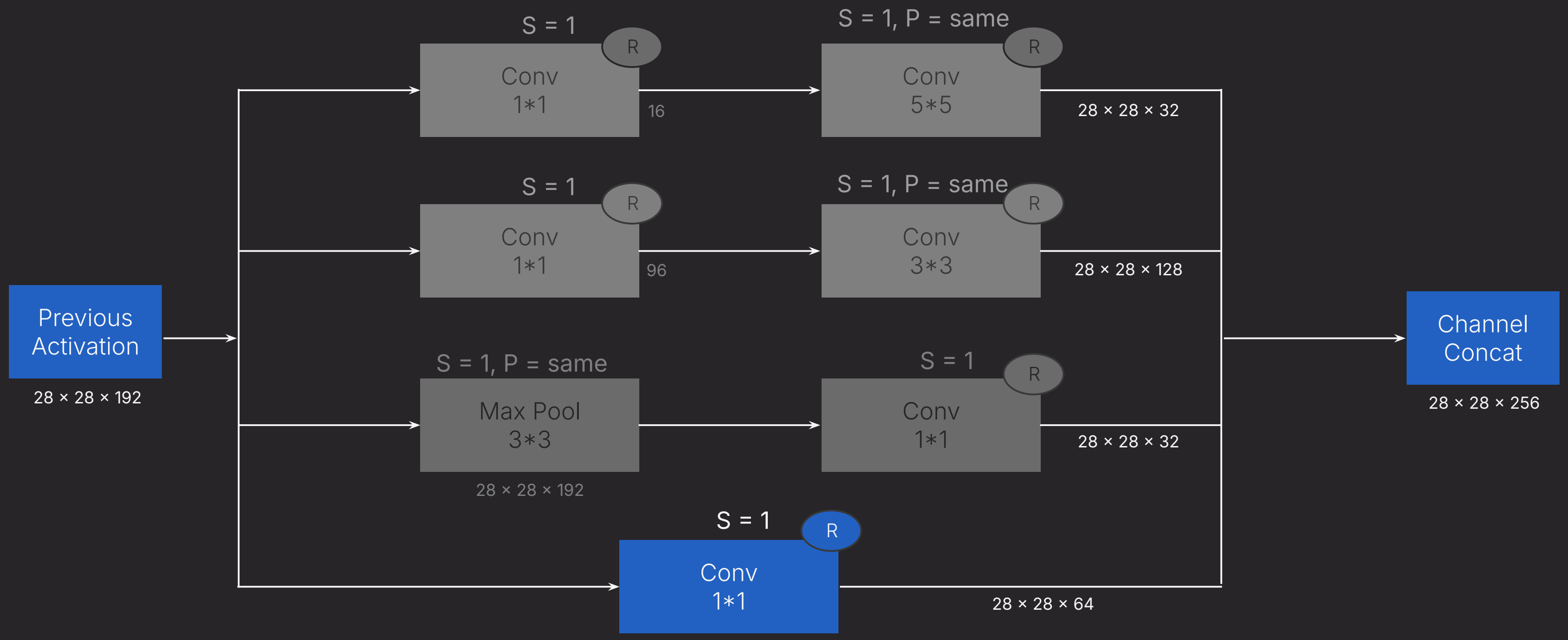
Inception Block

Why do we apply 1×1 convolution after max pooling ?

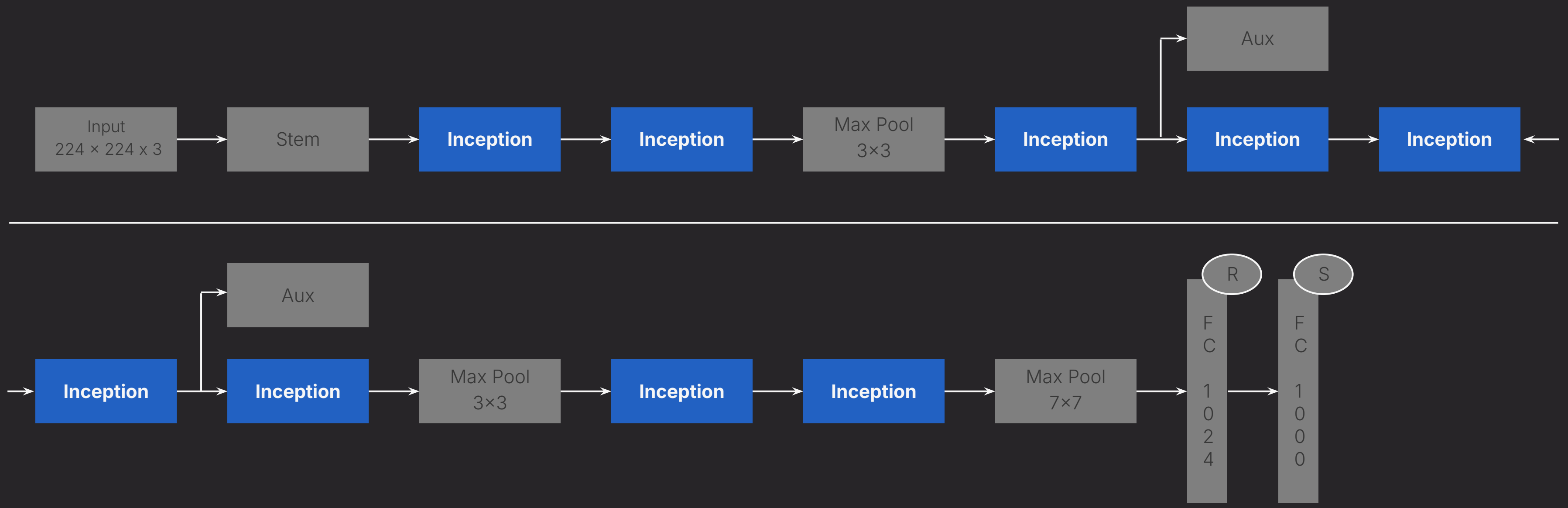


Inception Block

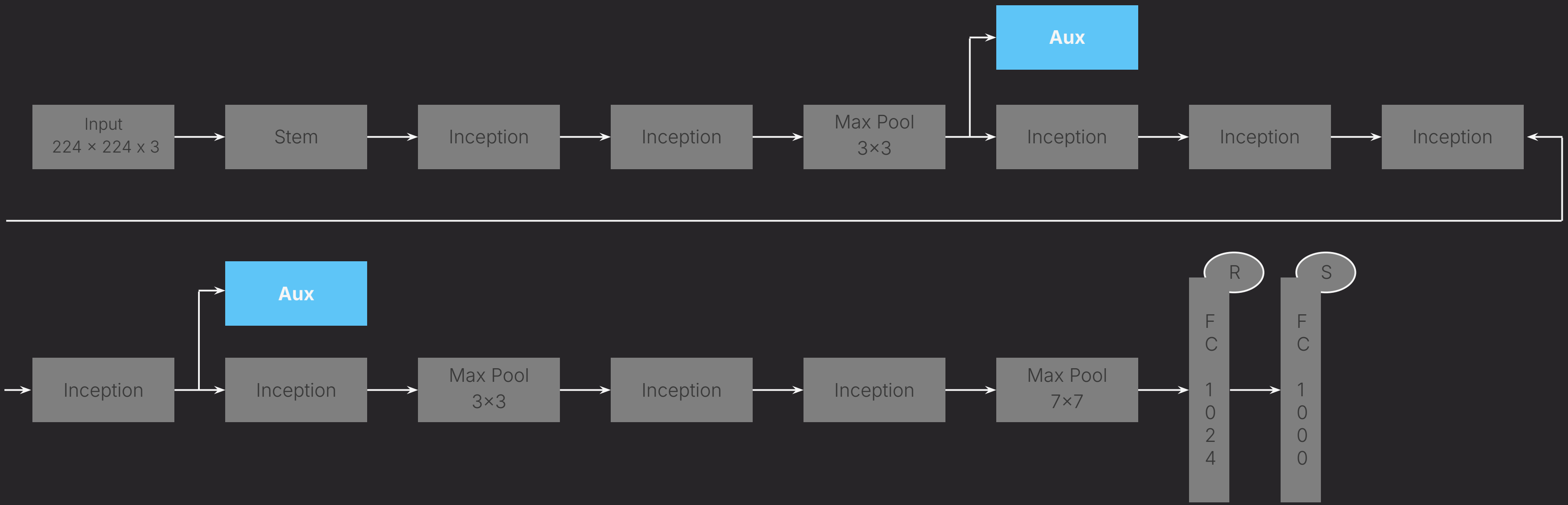
1×1 Convolution and Concatenation



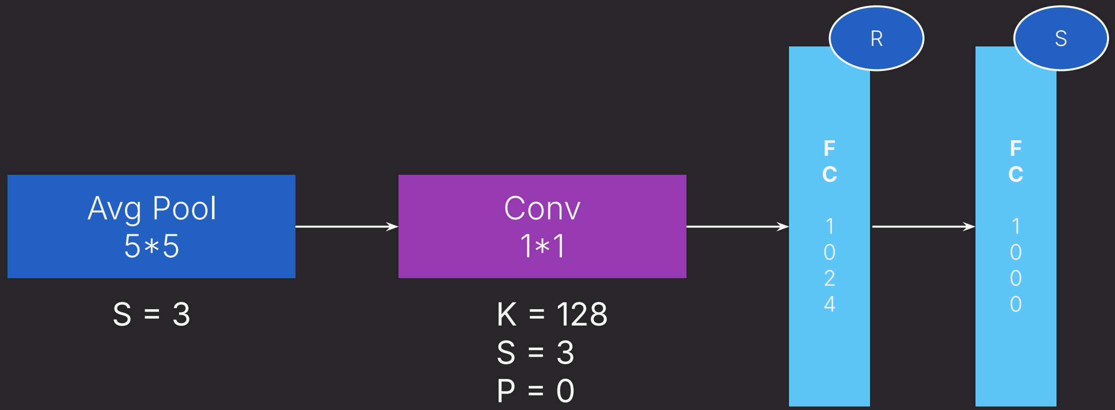
Inception Block



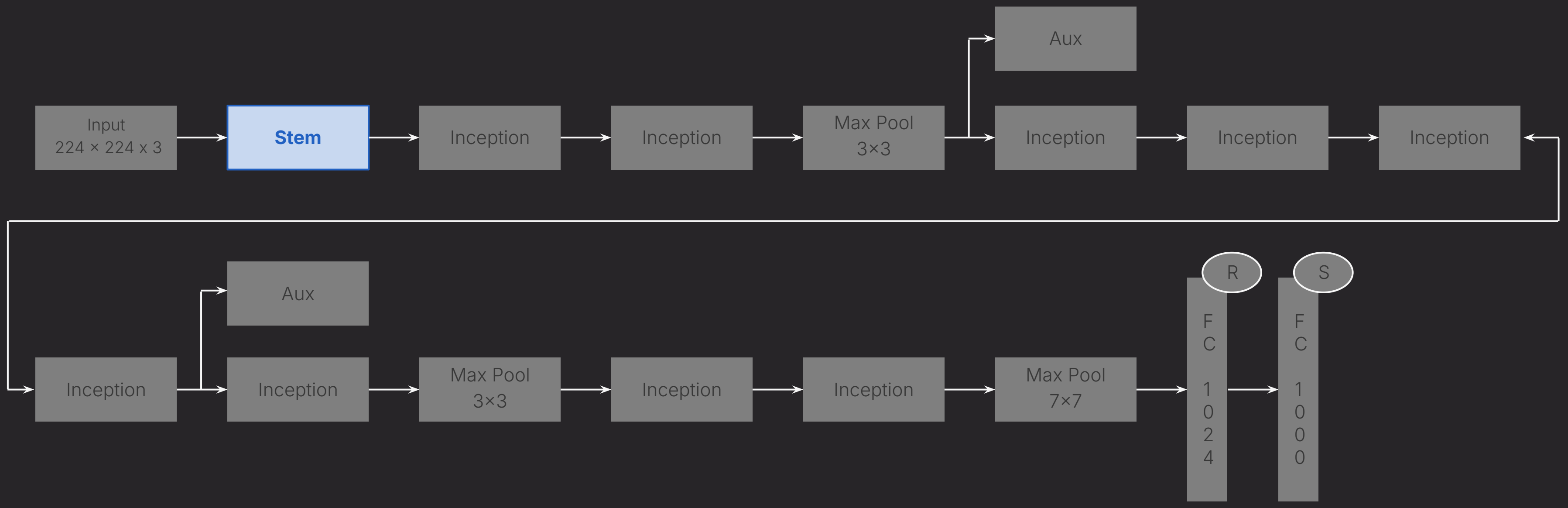
Auxiliary Classifier



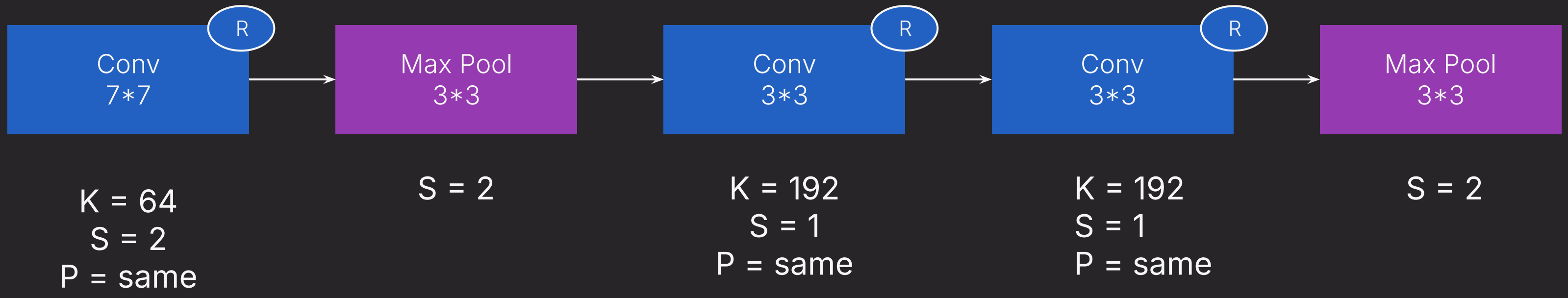
Auxiliary Classifier



GoogLeNet / Inception v1



The Stem Block



Evolution of the Inception Architecture

Inception-v2 (2015): Replacement of 5×5 convolutions with two sequential 3×3 convolutions to reduce in computational cost.

- Fewer parameters, leading to a more efficient model.

Inception-v3 (2016): Introduction of batch normalization and optimization in filter sizes across layers.

- Achieved state-of-the-art performance on ImageNet.

Inception-v4 and Introduction of Residual Connections: Addition of residual connections similar to those in ResNet.

- Enhanced efficiency and accuracy on ImageNet.

UpNext: Hands-on with Inception v1

What Made Inception v1 Good?

Inception v1 Efficiency:

- **Parameters:** Only 5 million parameters, compared to VGG16's 138 million.
- **1×1 Convolutions:** Reduces dimensionality, allowing for efficient use of resources.

Performance Achievements:

- **ILSVRC 2014:** Top-5 error rate of 6.67%, surpassing competitors like VGG16.
- **Adaptive Feature Capture:** Excels in diverse input scenarios by effectively capturing multi-scale features.

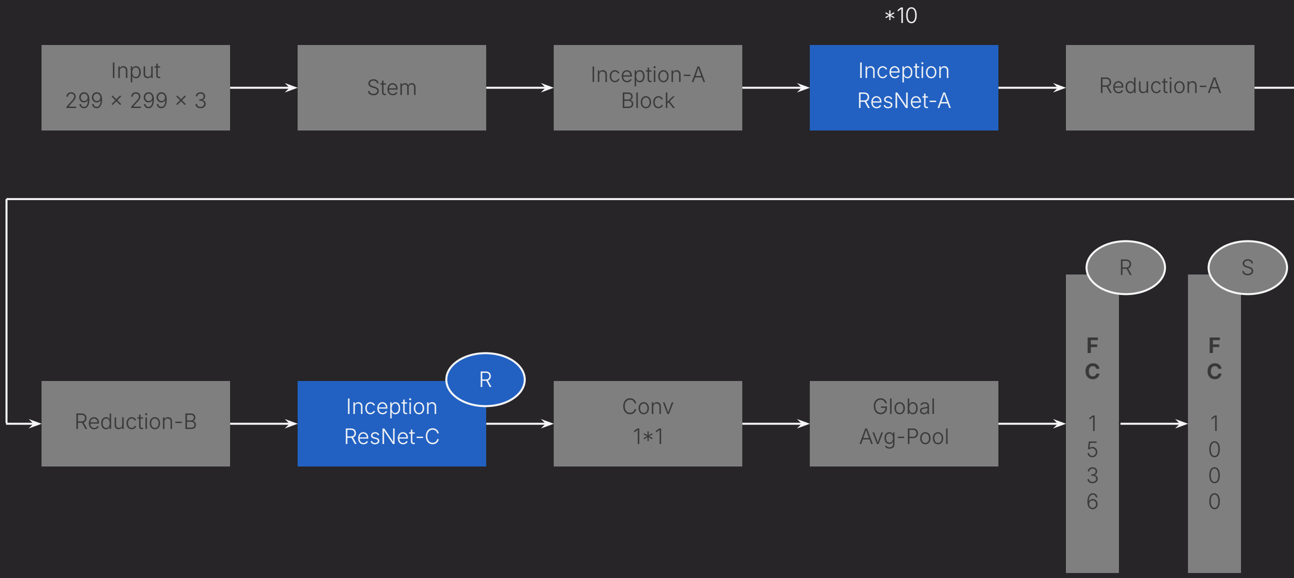
Limitation of Inception v1

Complexity and Computational Cost due to its deep and wide structure.

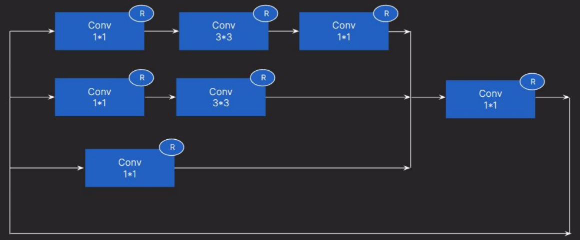
The Vanishing Gradients Problem was a common issue in deeper networks.

Fixed Grid Size Reduction might not have been optimal for all types of input sizes and shapes.

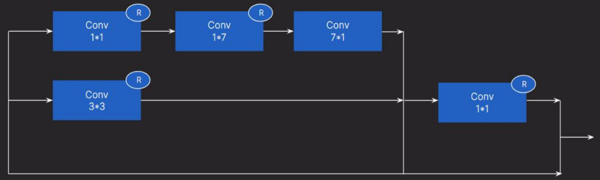
What Made Inception v2 Better?



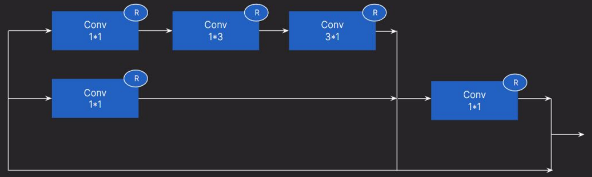
What Made Inception v2 Better?



Inception ResNet-A Block



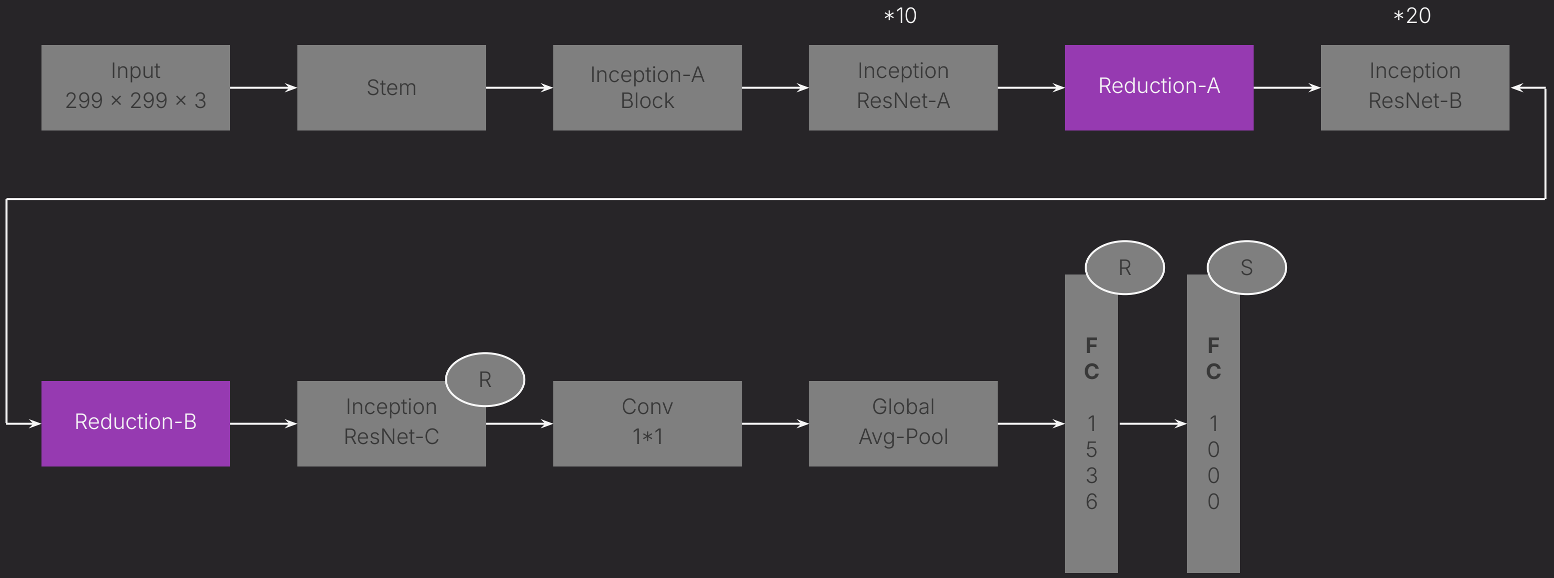
Inception ResNet-B Block



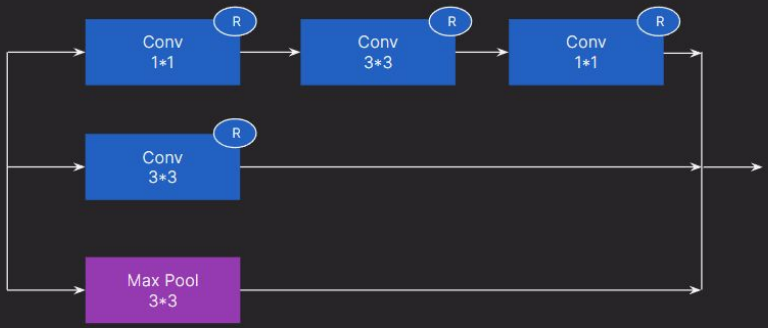
Inception ResNet-C Block

- **Inception-ResNet Modules:** Integrates ResNet's residual connections for deeper networks without vanishing gradient issues.

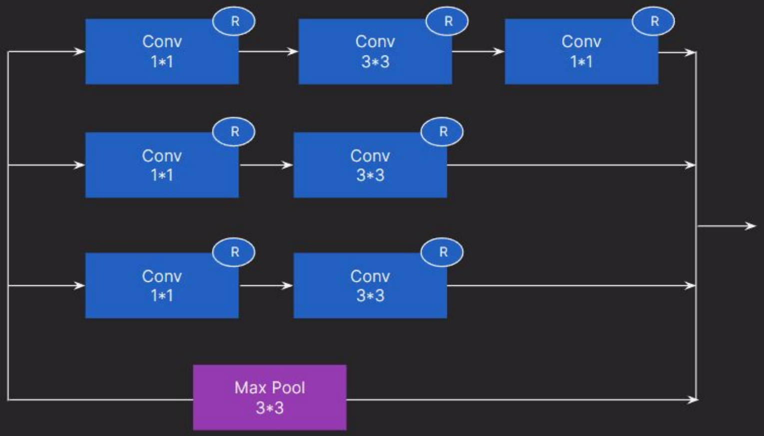
What Made Inception v2 Better?



What Made Inception v2 Better?

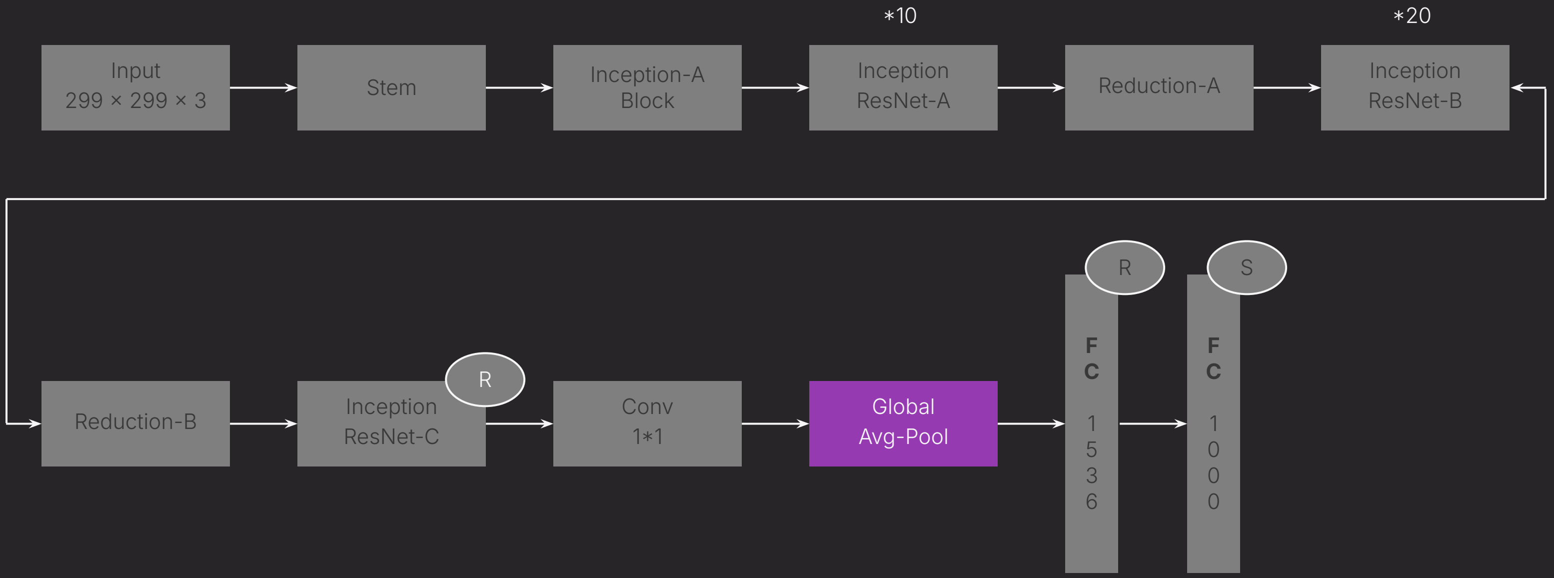


Reduction A Block



Reduction B Block

What Made Inception v2 Better?



Limitation of Inception v2

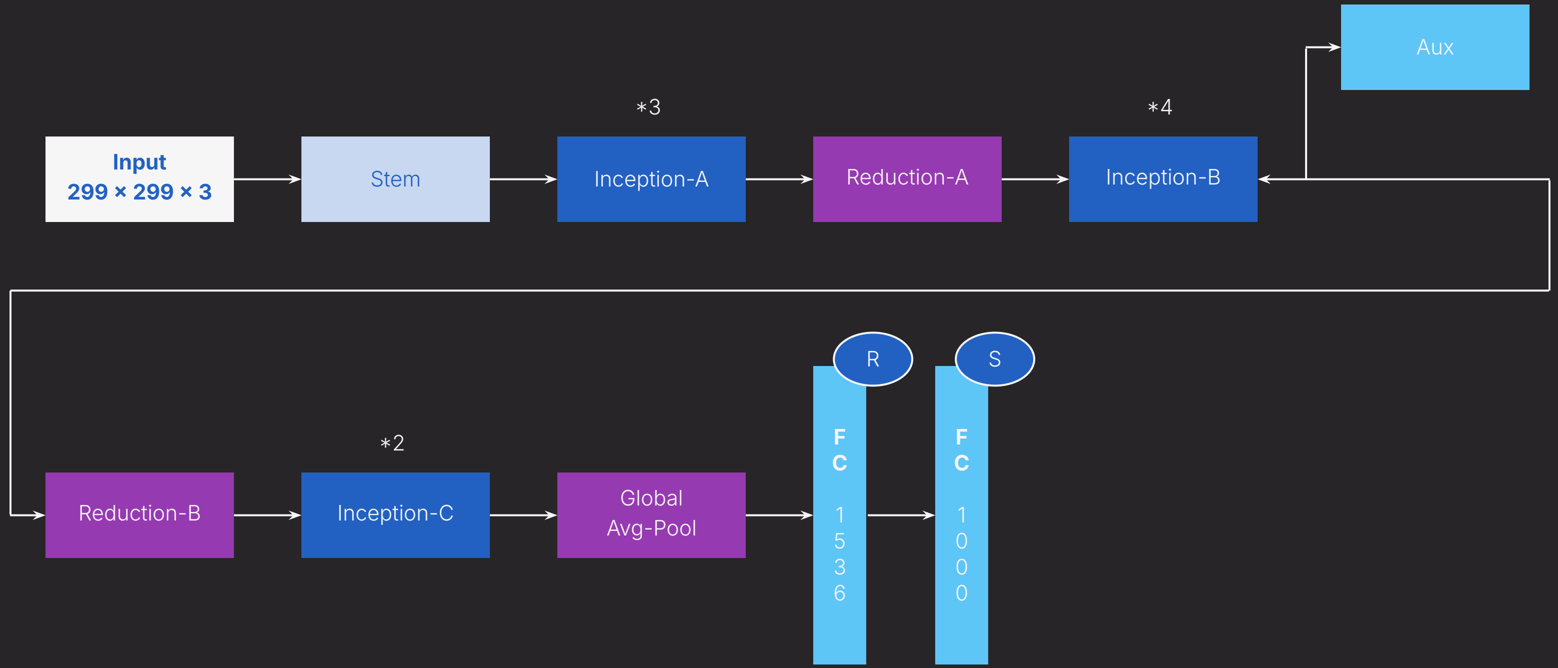
Parameter Efficiency: Optimization was necessary to reduce parameters and prevent resource-intensive models.

Handling Resolutions: Image sizes increased and adjustments were necessary to maintain performance.

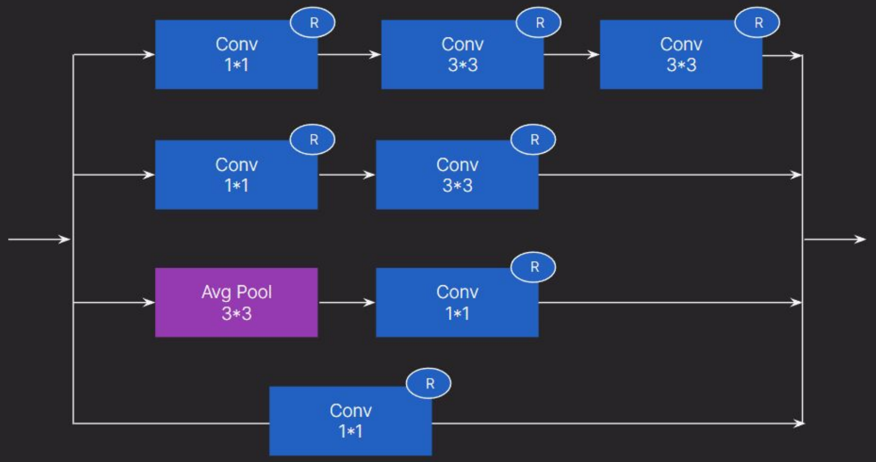
Scaling Challenges: Deeper layers didn't always improve performance due to optimization issues.

Inception v3

- Developed in the year 2015.



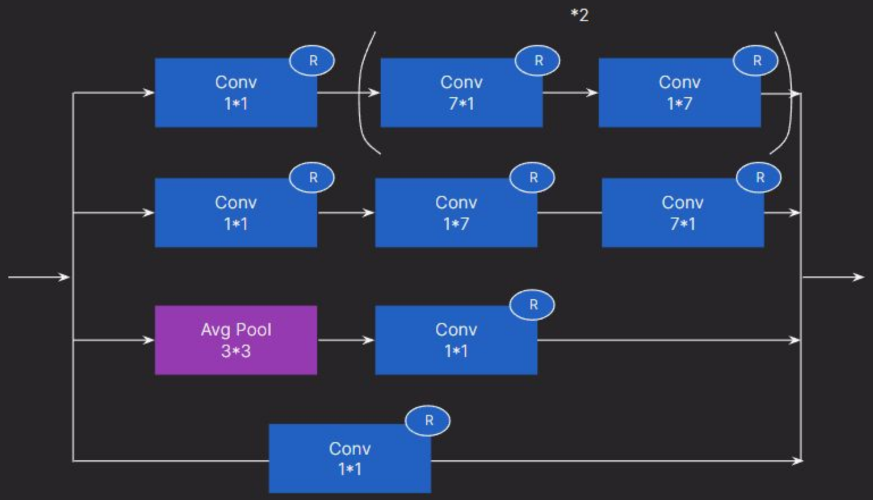
Inception v3: Architecture



Inception A Block

The Inception A Block aims to **expand the network's capacity to capture complex patterns** through convolutional operations without much reduction in spacial dimensions.

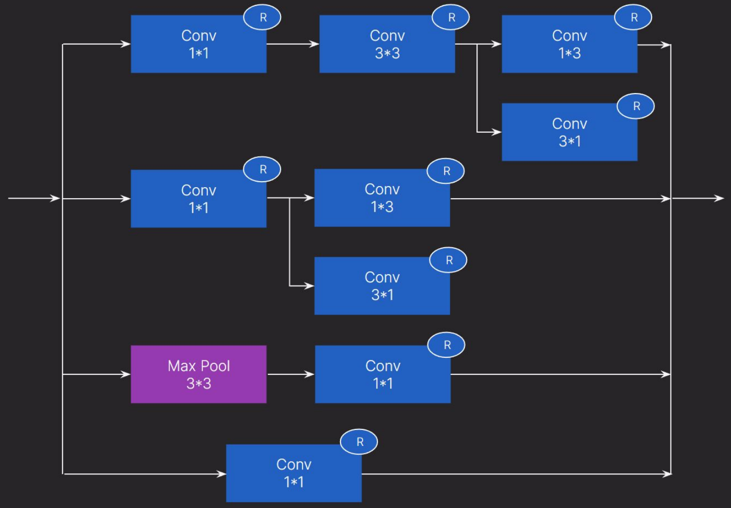
Inception v3: Architecture



Inception B Block

- The Inception B Block aims to **reduce spacial dimensions more significantly than Block A**.
- It also leverages **large strides (1*7)** and **asymmetric factorizations** to reduce computational costs.

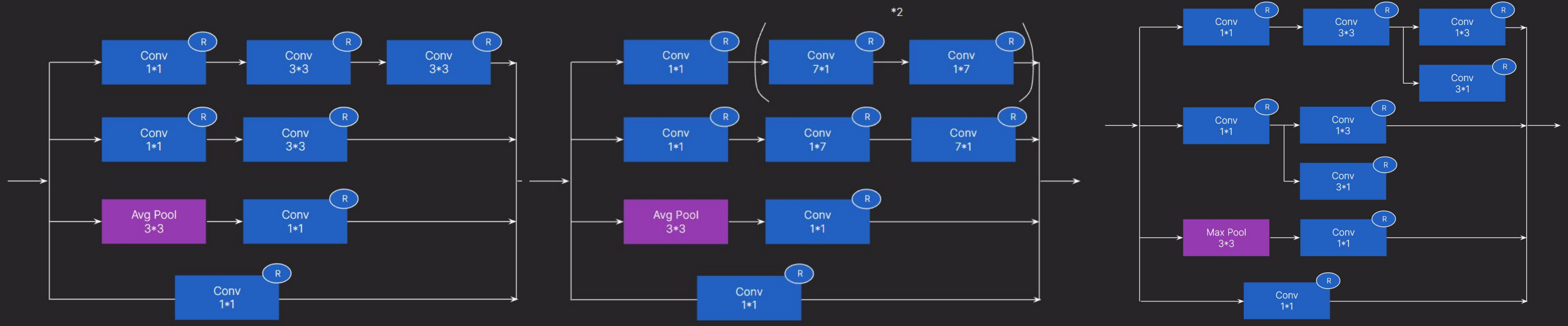
Inception v3: Architecture



Inception C Block

- The Inception C Block focuses on **high level feature extraction and processing**.
- It intensifies the use of asymmetric factorizations to maximize efficiency and feature extraction.

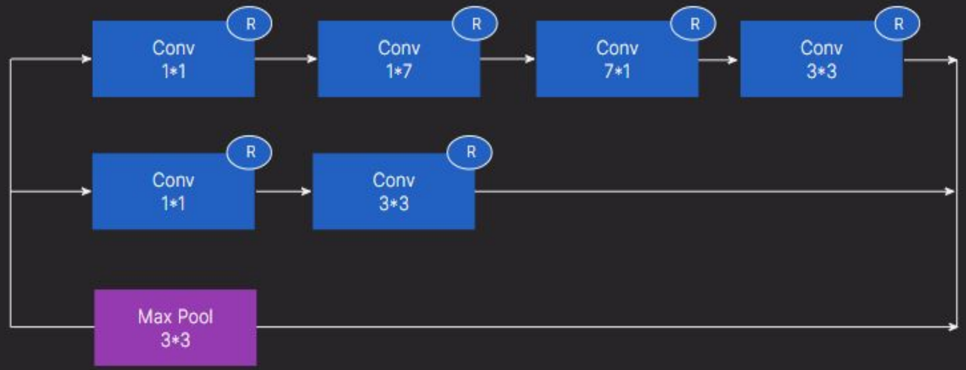
Inception v3: Architecture



Inception A Block **Inception B Block** **Inception C Block**

- All 3 block use batch normalization and ReLU activation functions after each convolution is complete.

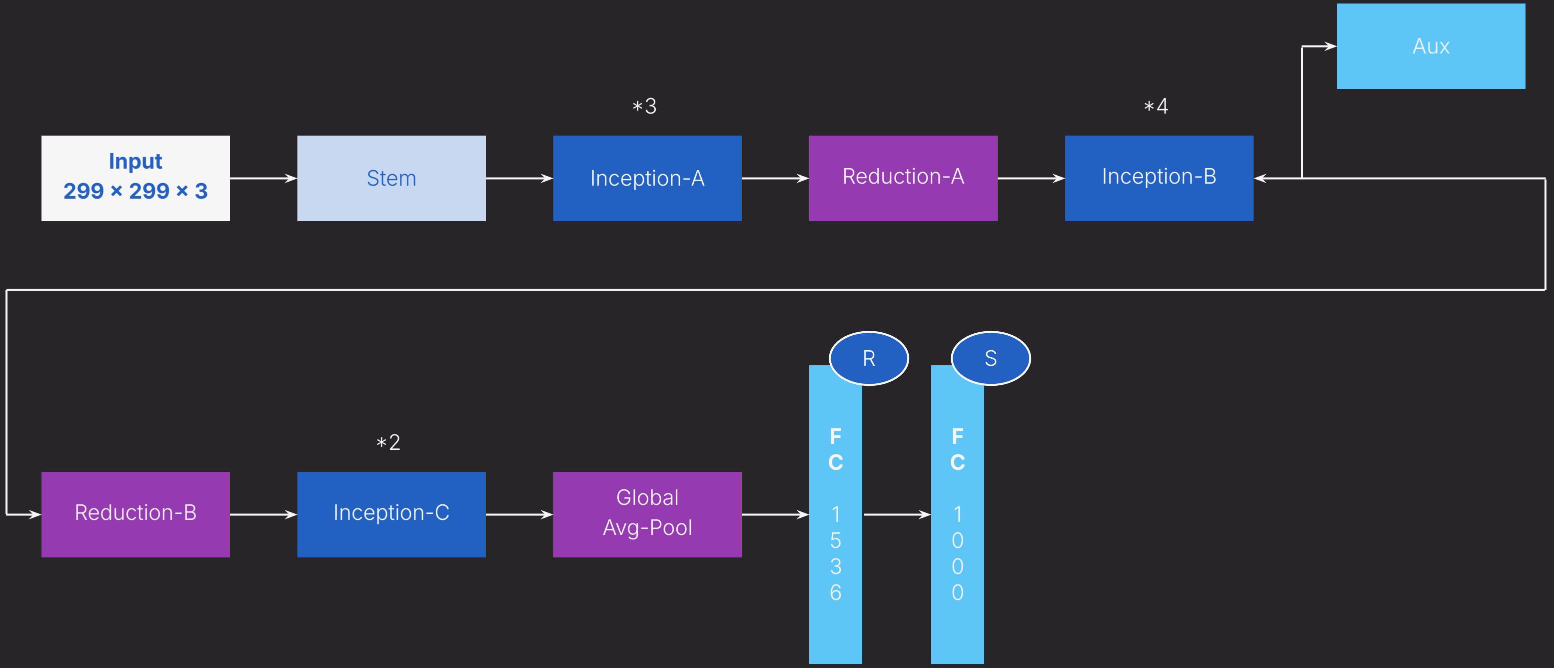
Inception v3: Architecture



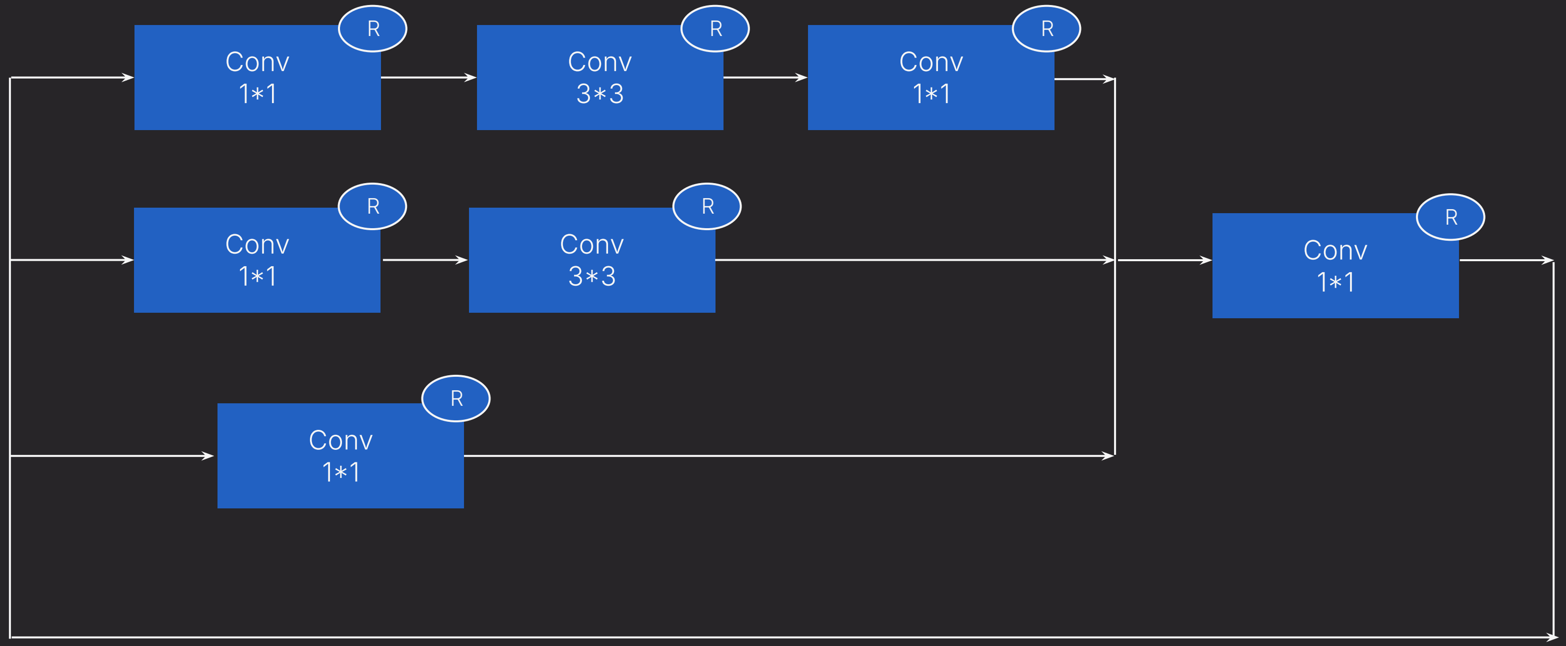
Reduction B Block

- Inception V3 incorporates **asymmetric convolutions**, enhancing complexity and efficiency.
- V3's convolutions **delicately adjust to spatial characteristics**, optimizing feature map handling.

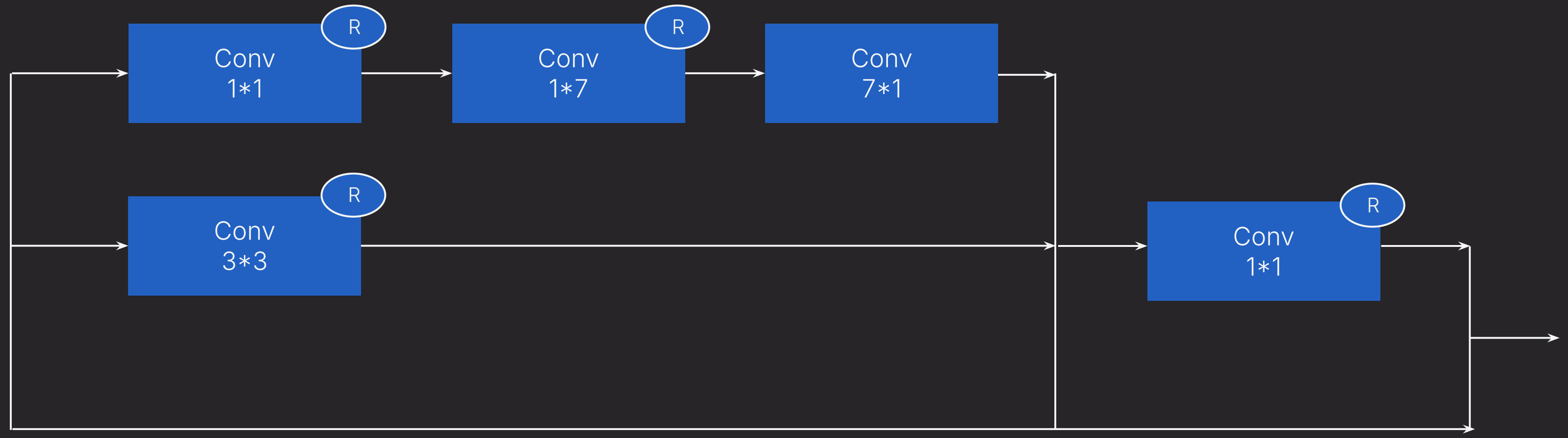
Inception v3: Architecture



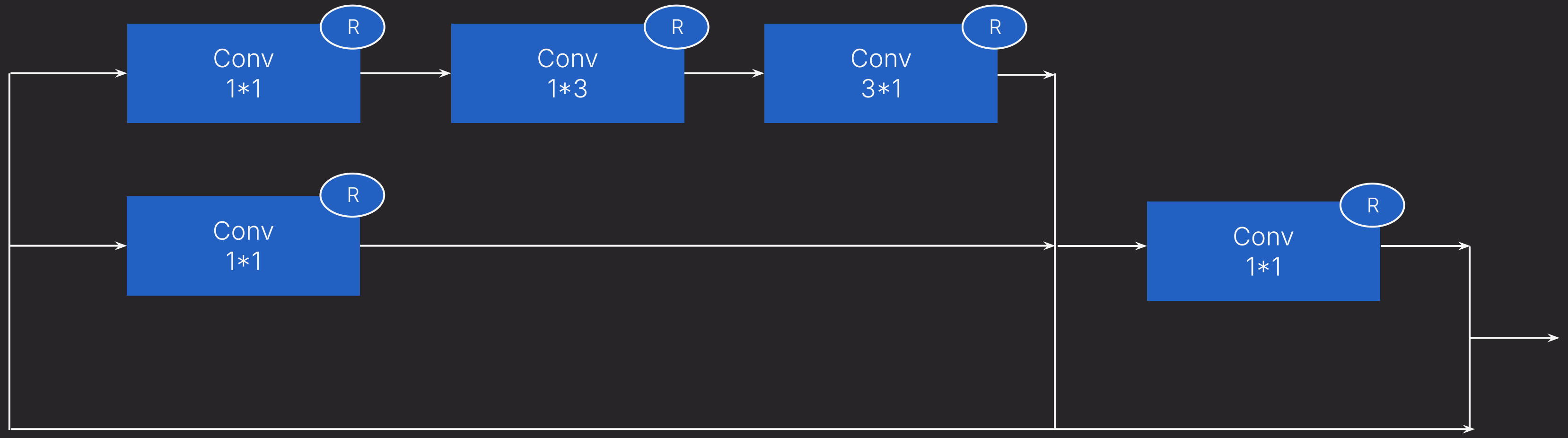
Resnet A block



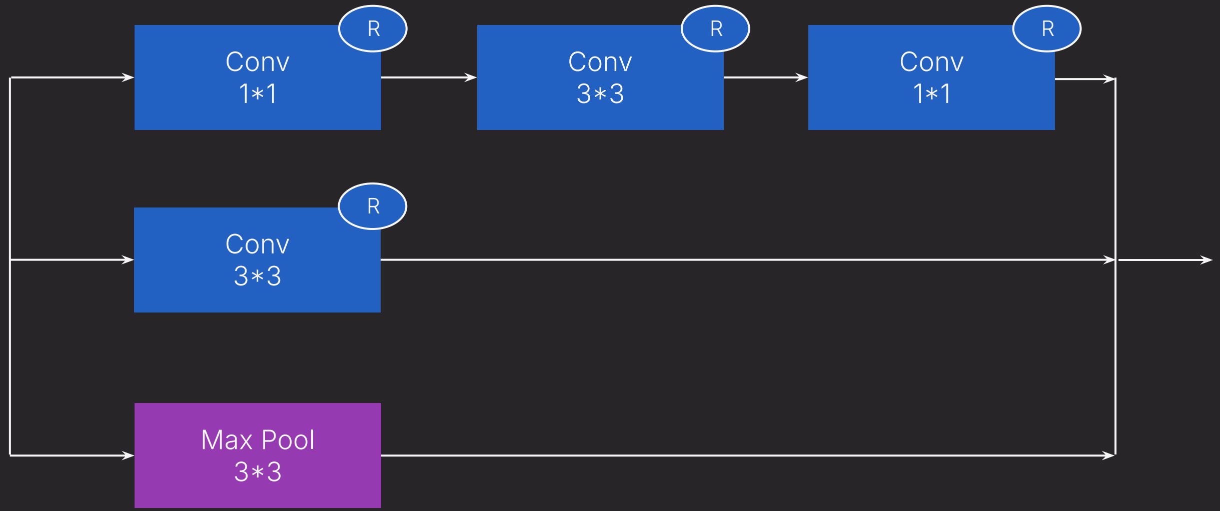
Resnet b block



Resnet c block

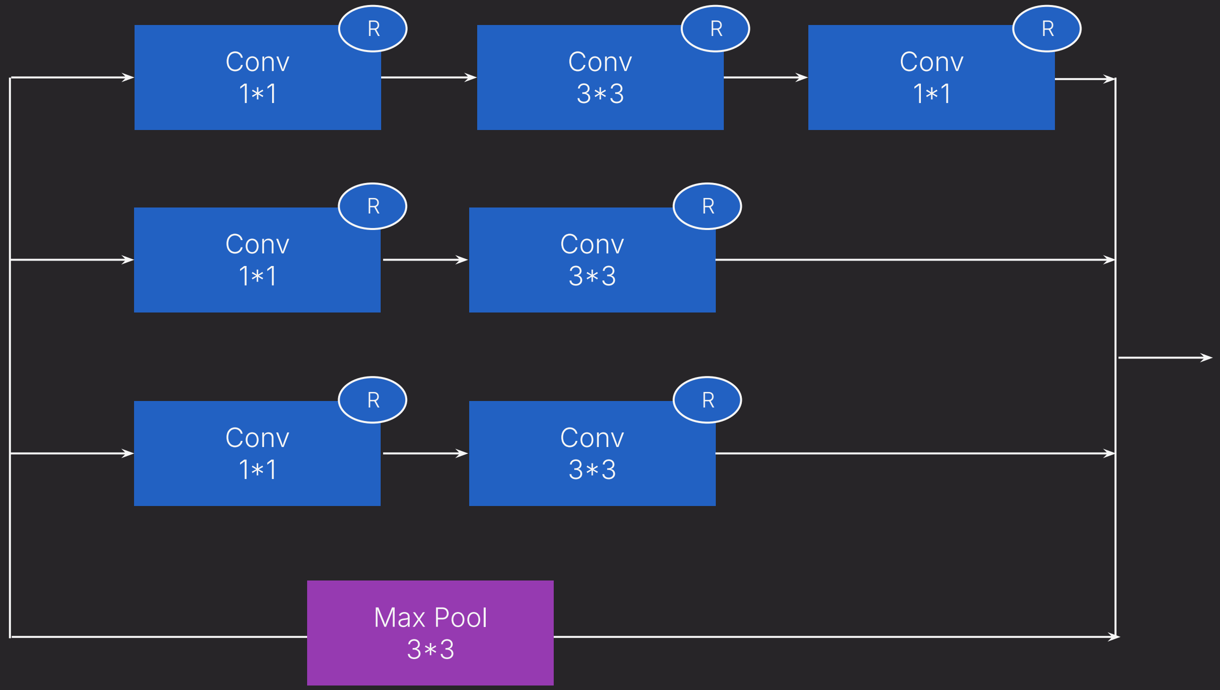


Reduction A block

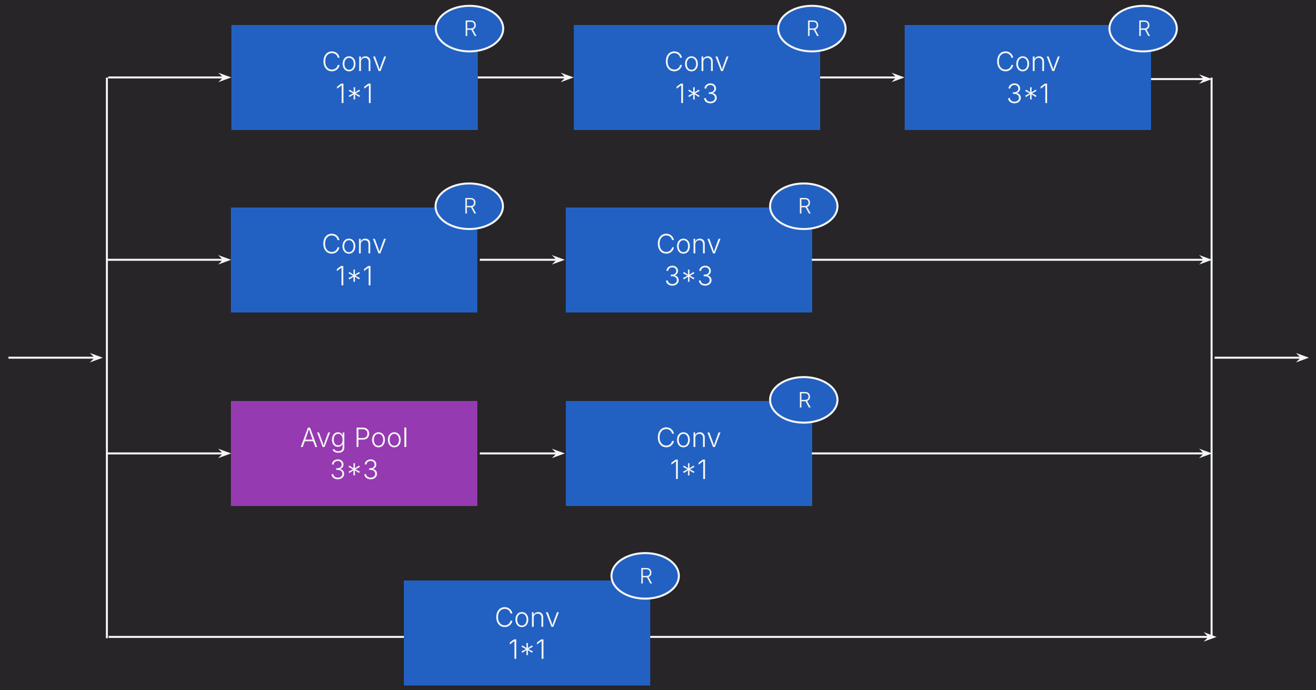


Reduction b block

Analytics Vidhya

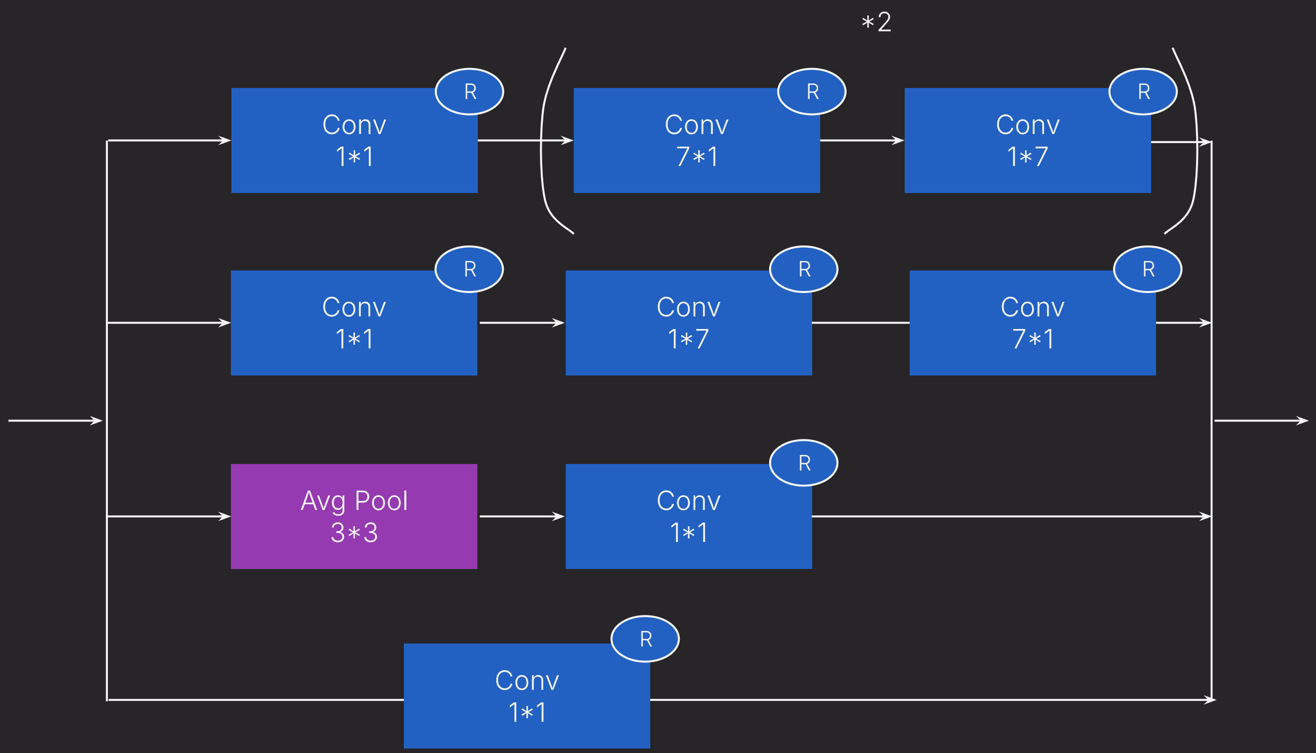


Inception v3 A block

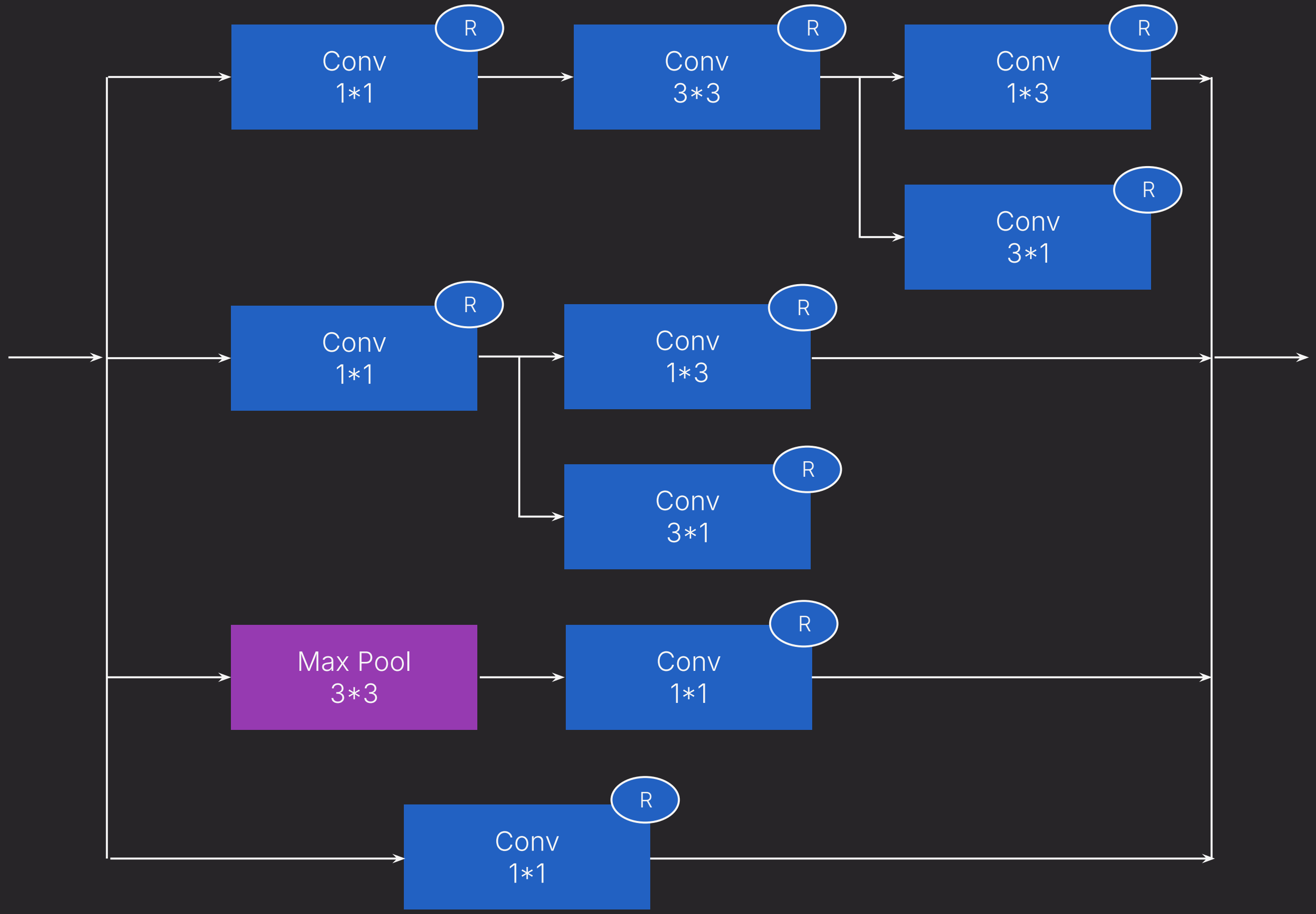


Analytics Vidhya

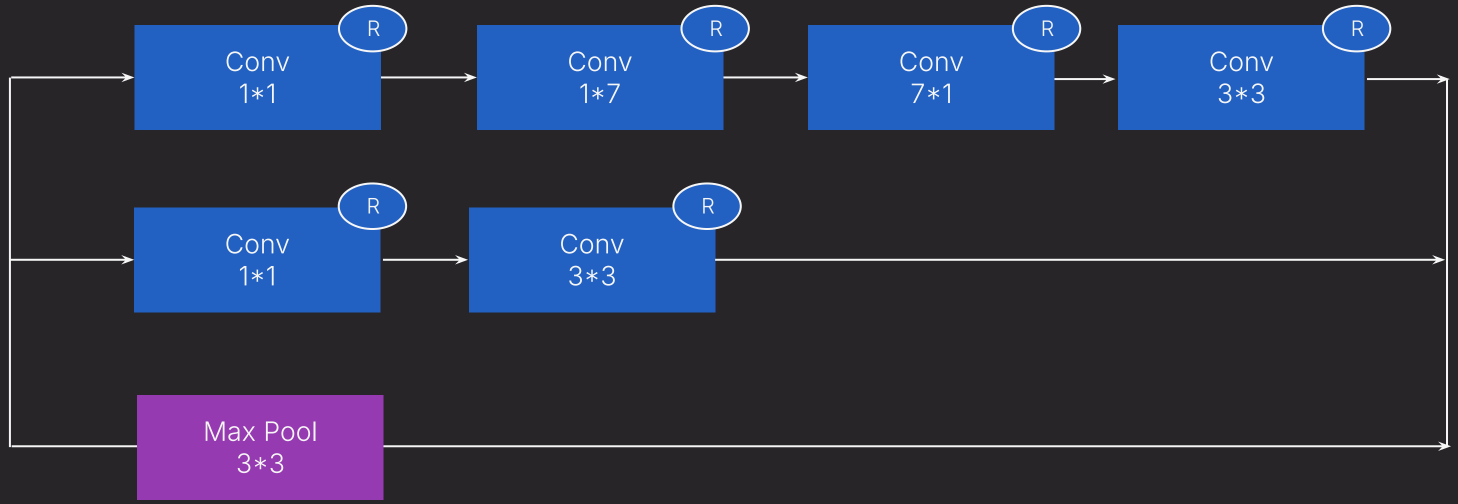
Inception v3 B block

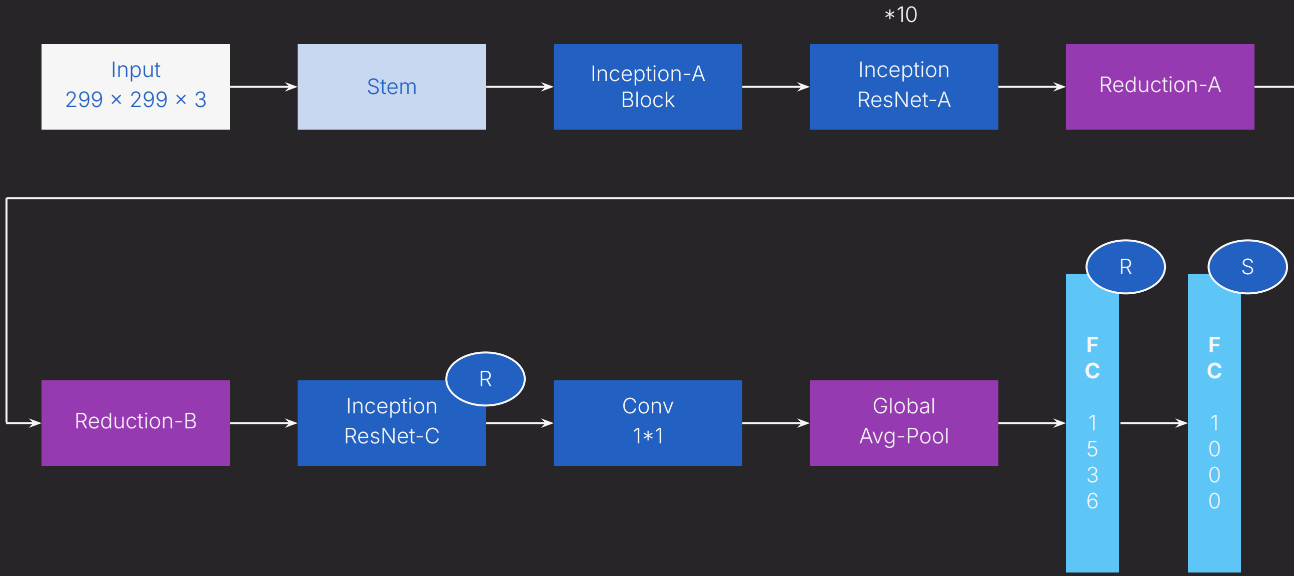


Inception v3 C block

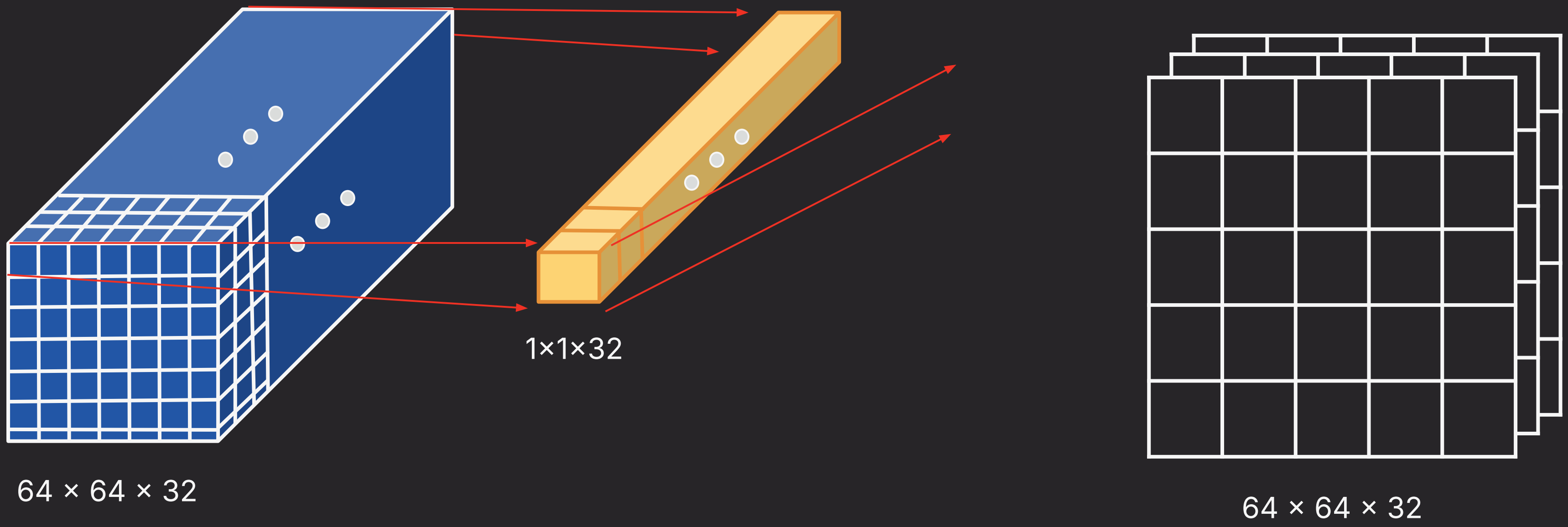


Reduction B block

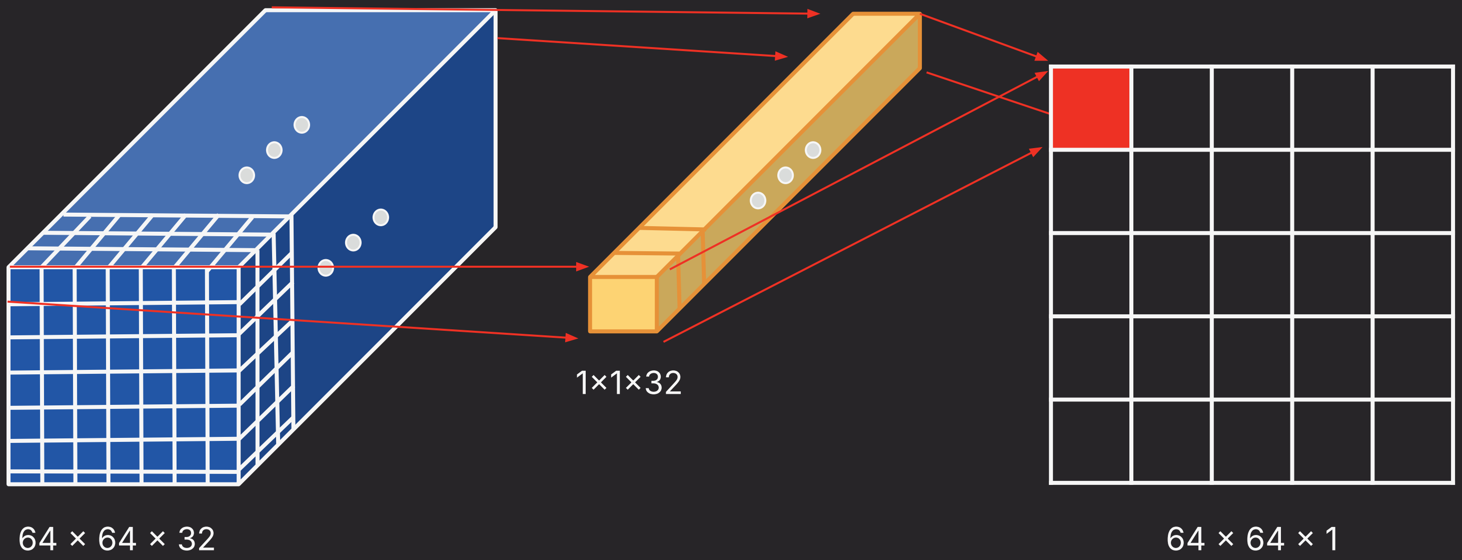



*10
*20
Input
299 × 299 × 3
Stem
Inception-A
Block
Inception
ResNet-A
Reduction-A
Inception
ResNet-B
Reduction-B
Inception
ResNet-C
R
Conv
1*1
Global
Avg-Pool
R
FC
1536
S
FC
1000

Inception Block



Inception Block



Inception Block