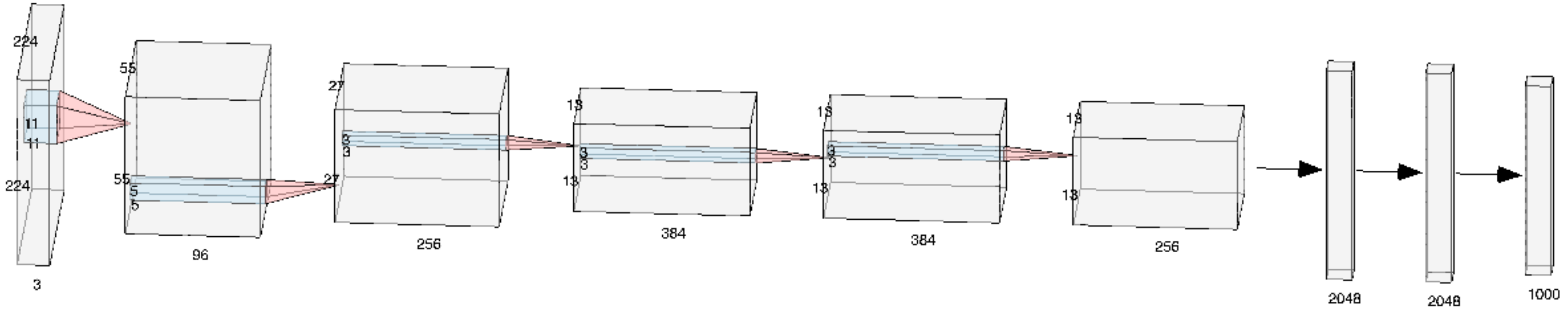
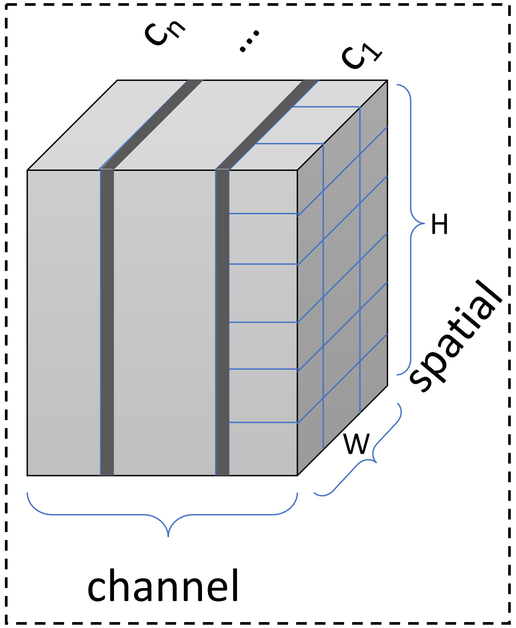
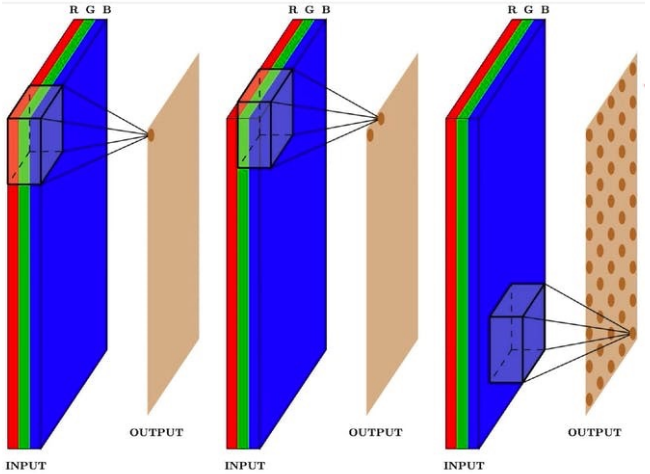
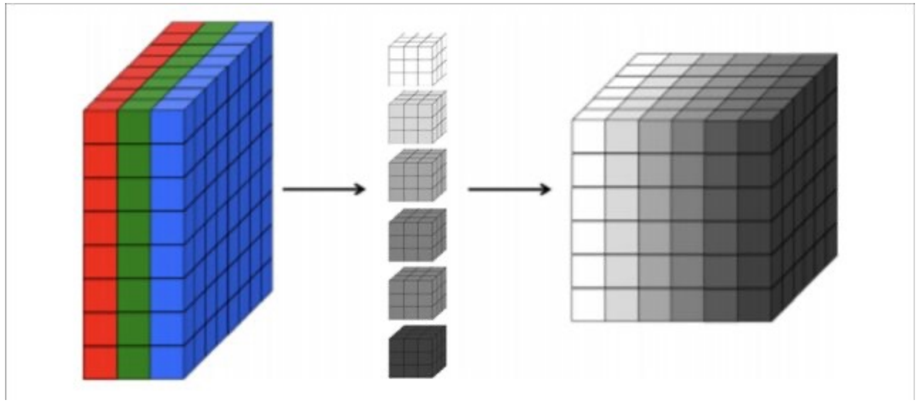
SENet(Squeeze and Excitation Networks)





- CNN extracts informative features by fusing spatial and channel-wise information together within local receptive fields

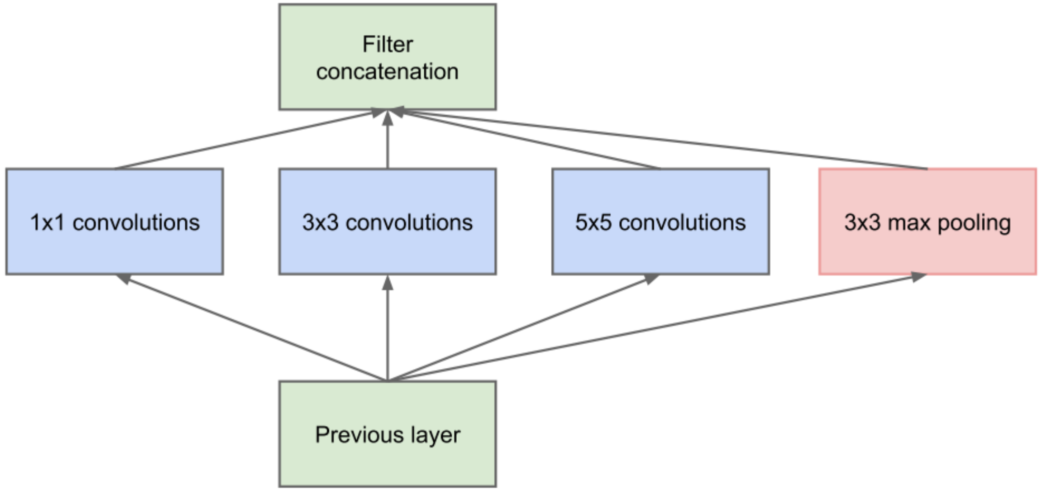


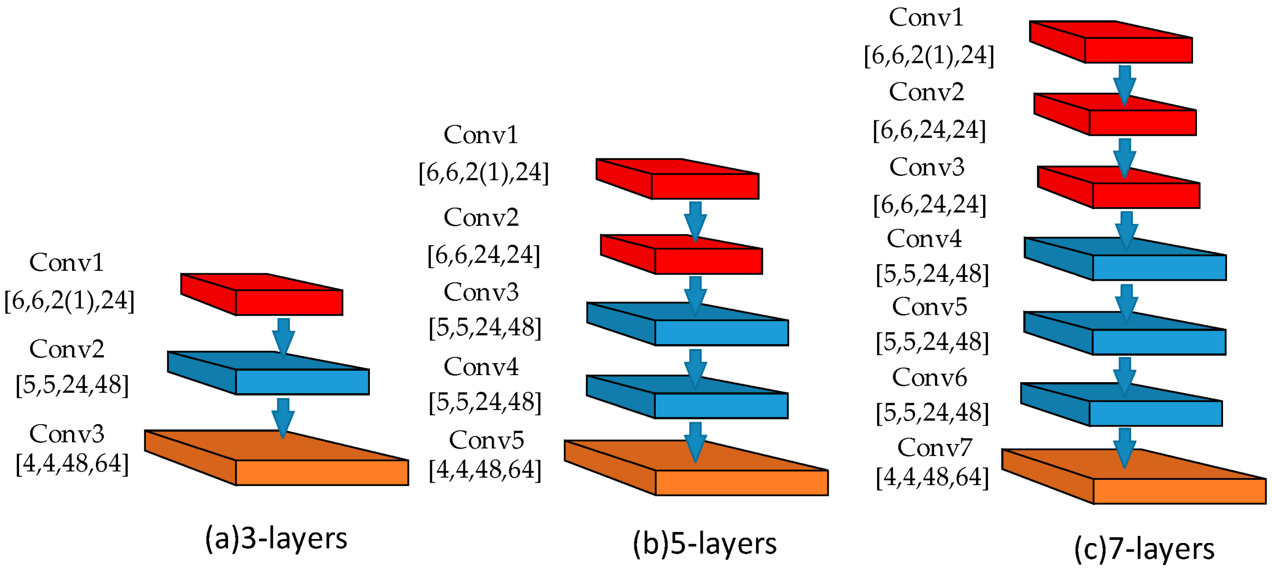


- For each convolutional layer, a set of filters are learned to express local spatial connectivity patterns along input channels

Introduction

- By stacking a series of convolutional layers interleaved with non-linearities and downsampling, CNNs are capable of capturing hierarchical patterns with global receptive fields as powerful image descriptions.
- **Inception** architectures (***GoogleNet***), which showed that the network can achieve competitive accuracy by embedding multi-scale processes in its modules





Introduction

- *In this paper, we investigate a different aspect of architectural design - the channel relationship, by introducing a new architectural unit, which we term the "Squeeze-and Excitation" (SE) block. Our goal is to improve the representational power of a network by explicitly modelling the interdependencies between the channels of its convolutional features*
- All of this works by fusing the spatial and channel information of an image. The different filters will first find spatial features in each input channel before adding the information across all available output channels.

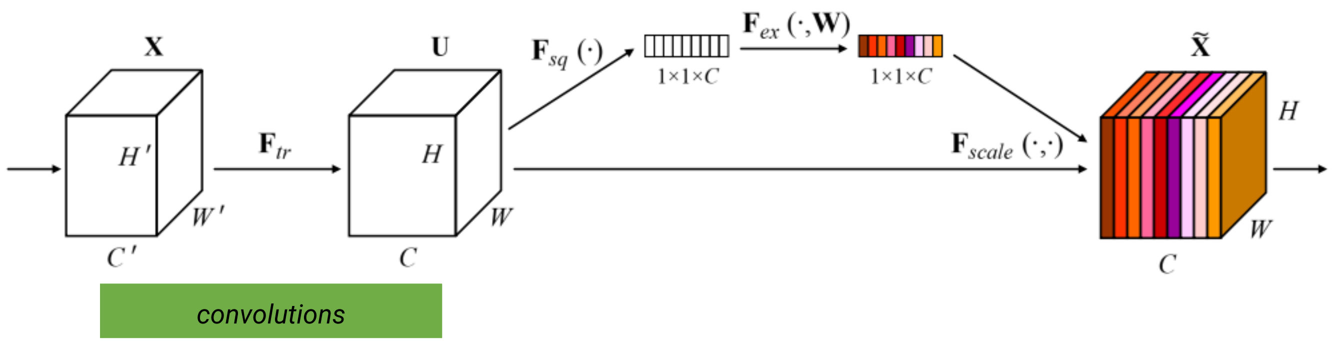
A Siamese Neural Network for Non-Invasive Baggage Re-Identification

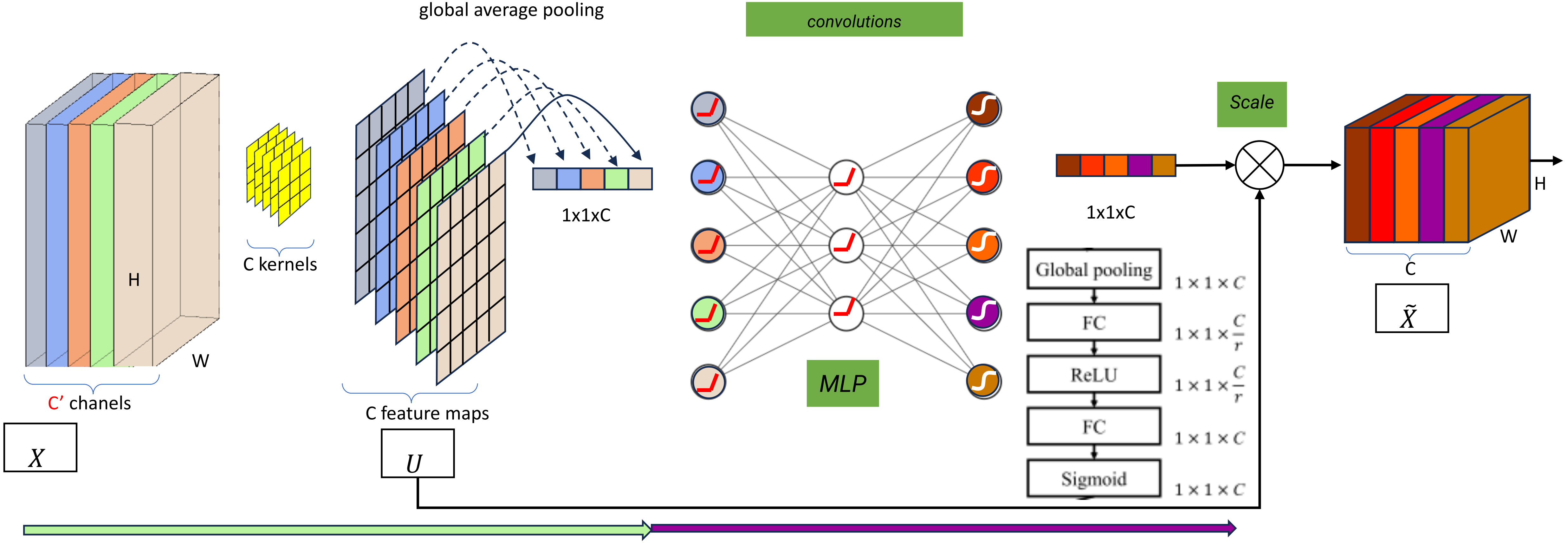
by Pier Luigi Mazzeo [1], Christian Libetta [2], Paolo Spagnolo [1,*] and Cosimo Distante [1]

SE (Squeeze and Excitation) Block

For any given transformation

$$F_{tr}: X \longrightarrow U, X \in R^{H'W'C'}, U \in R^{HWC}$$



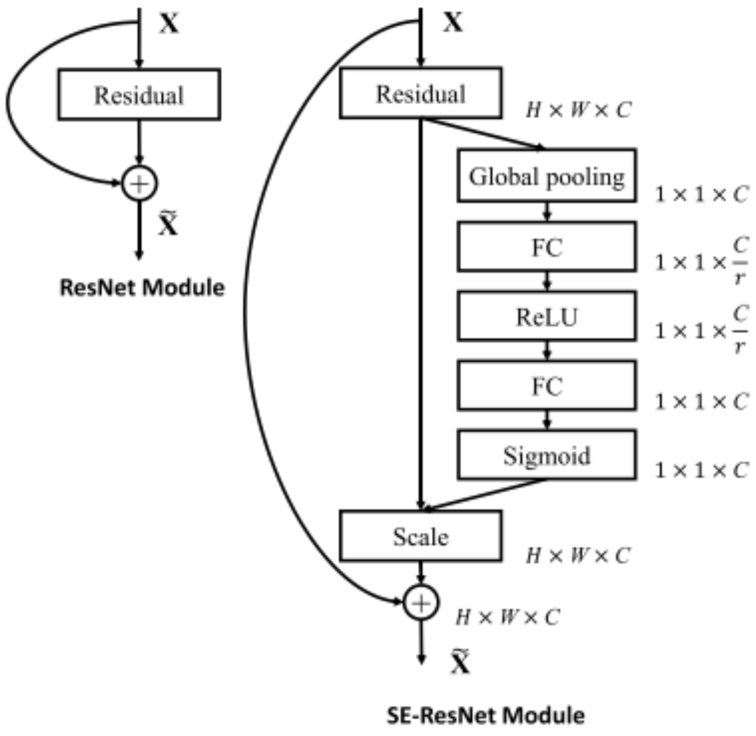


Squeeze
aggregates the feature maps

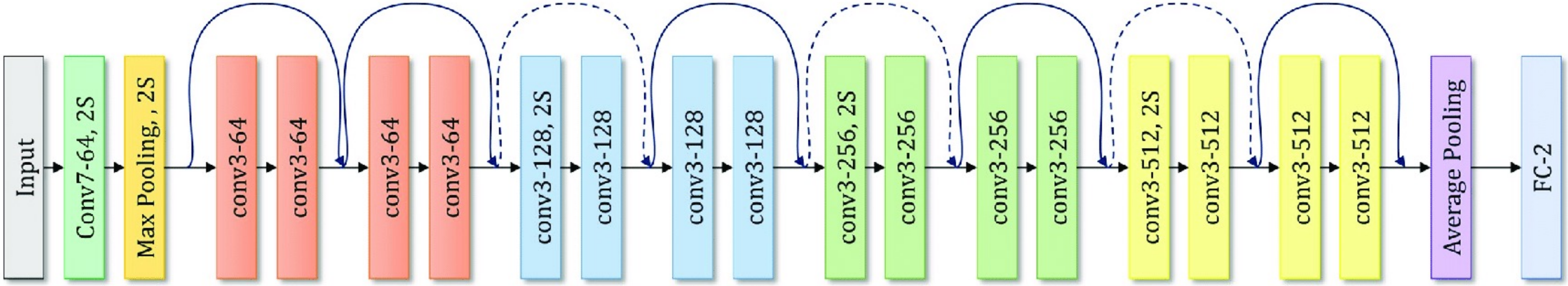
Excitation
learned importance weights

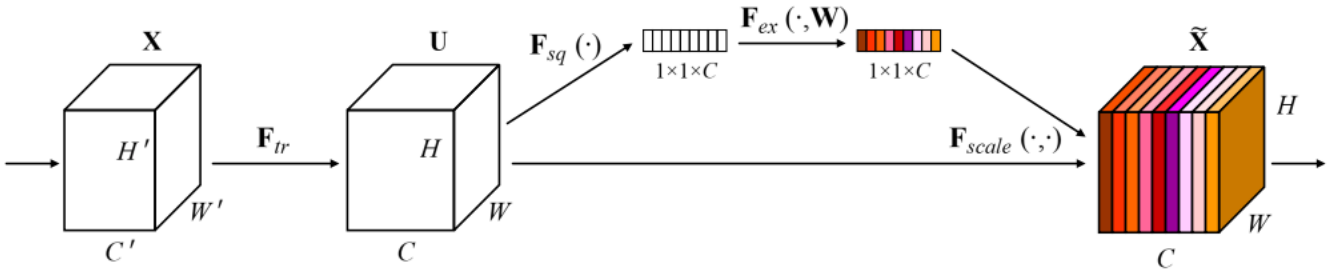
SEBlock

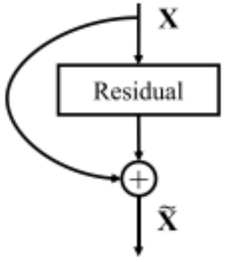
```python
class SEBlock(nn.Module):
    def __init__(self, input_channels, reduction_ratio=16):
        super(SEBlock, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.fc1 = nn.Linear(input_channels, input_channels // reduction_ratio, bias=False)
        self.relu = nn.ReLU(inplace=True)
        self.fc2 = nn.Linear(input_channels // reduction_ratio, input_channels, bias=False)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        batch_size, channels, _, _ = x.size()
        y = self.avg_pool(x).view(batch_size, channels)
        y = self.fc1(y)
        y = self.relu(y)
        y = self.fc2(y)
        y = self.sigmoid(y).view(batch_size, channels, 1, 1)
        return x * y.expand_as(x)
```



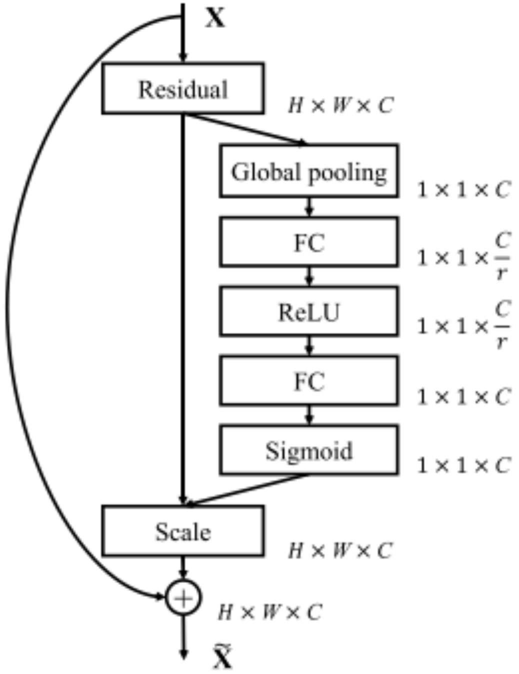
Implementation: Resnet18 - SENetwork







ResNet Module

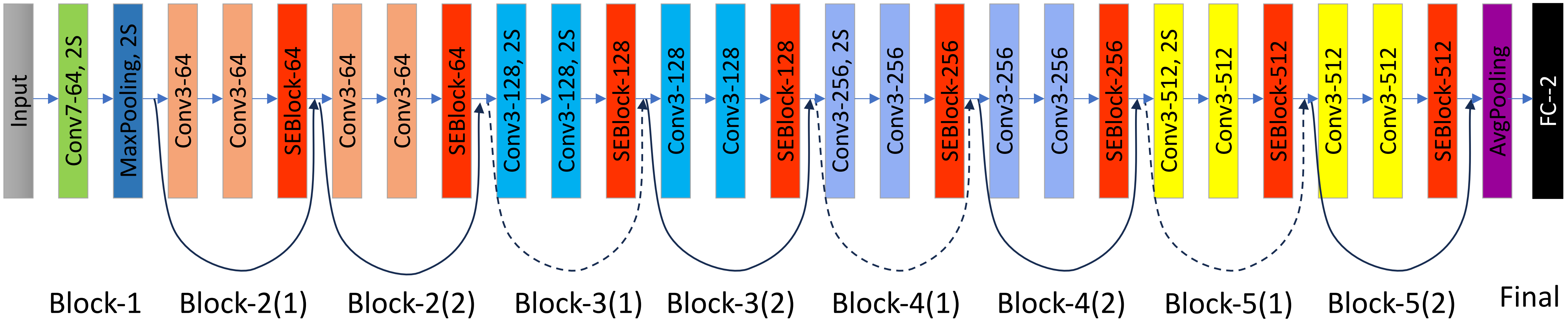


SE-ResNet Module

- To integrate SE blocks into the ResNet18 model:

1. Define the SEBlock.
2. Add SEBlock instances after each block of the ResNet18 architecture

SEResNet18


Input
Conv7-64, 2S
MaxPooling, 2S
Conv3-64
Conv3-64
SEBlock-64
Conv3-64
Conv3-64
SEBlock-64
Conv3-128, 2S
Conv3-128, 2S
SEBlock-128
Conv3-128
Conv3-128
SEBlock-128
Conv3-256, 2S
Conv3-256
SEBlock-256
Conv3-256
Conv3-256
SEBlock-256
Conv3-512, 2S
Conv3-512
SEBlock-512
Conv3-512
Conv3-512
SEBlock-512
AvgPooling
FC--2
Block-1
Block-2(1)
Block-2(2)
Block-3(1)
Block-3(2)
Block-4(1)
Block-4(2)
Block-5(1)
Block-5(2)
Final

SEResnet18

- **E:\thanh\ntu_group\thinh\Image-Classification\SEResnet18\train_se_resnet18.py**
- Training: 100%|██████████| 40/40 [00:35<00:00, 1.14it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.04it/s]
- Epoch 0: Validation loss = 1.593268911043803, Validation accuracy: 0.5
- Validation accuracy increased (0 --> 0.5). Model saved
- Training: 100%|██████████| 40/40 [00:34<00:00, 1.15it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.04it/s]
- Epoch 1: Validation loss = 0.7448241660992304, Validation accuracy: 0.6537295778592428
- Validation accuracy increased (0.5 --> 0.6537295778592428). Model saved
- Training: 100%|██████████| 40/40 [00:34<00:00, 1.15it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.04it/s]
- Epoch 2: Validation loss = 0.8086711913347244, Validation accuracy: 0.6246566226085027
- Training: 100%|██████████| 40/40 [00:38<00:00, 1.03it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.05it/s]
- Epoch 3: Validation loss = 0.5508396079142889, Validation accuracy: 0.7178877095381418
- Validation accuracy increased (0.6537295778592428 --> 0.7178877095381418). Model saved
- Training: 100%|██████████| 40/40 [00:49<00:00, 1.25s/it]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.05it/s]
- Epoch 4: Validation loss = 0.9465955793857574, Validation accuracy: 0.6052251011133194
- Training: 100%|██████████| 40/40 [00:35<00:00, 1.14it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.06it/s]
- Epoch 5: Validation loss = 0.4808992240577936, Validation accuracy: 0.8208229939142863
- Validation accuracy increased (0.7178877095381418 --> 0.8208229939142863). Model saved
- Training: 100%|██████████| 40/40 [00:34<00:00, 1.14it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.02it/s]
- Epoch 6: Validation loss = 1.2058287014563878, Validation accuracy: 0.5446381519238154
- Training: 100%|██████████| 40/40 [00:35<00:00, 1.14it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.05it/s]
- **Epoch 7: Validation loss = 0.33514292041460675, Validation accuracy: 0.8705415378014246**
- Validation accuracy increased (0.8208229939142863 --> 0.8705415378014246). Model saved
- Training: 100%|██████████| 40/40 [00:35<00:00, 1.14it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.03it/s]
- Epoch 8: Validation loss = 0.7108310361703237, Validation accuracy: 0.7303647299607595
- Training: 100%|██████████| 40/40 [00:35<00:00, 1.14it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.04it/s]
- Epoch 9: Validation loss = 0.3489784523844719, Validation accuracy: 0.8559288581212362

- Process finished with exit code 0

Resnet18

- **E:\thanh\ntu_group\thinh\Image-Classification\SEResnet18\train_resnet18.py**
- Training: 100%|██████████| 40/40 [01:31<00:00, 2.28s/it]
- Valid: 100%|██████████| 12/12 [00:09<00:00, 1.28it/s]
- Epoch 0: Validation loss = 1.7880469659964244, Validation accuracy: 0.5
- Validation accuracy increased (0 --> 0.5). Model saved
- Training: 100%|██████████| 40/40 [00:34<00:00, 1.17it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.12it/s]
- Epoch 1: Validation loss = 0.6179960519075394, Validation accuracy: 0.7065854867299398
- Validation accuracy increased (0.5 --> 0.7065854867299398). Model saved
- Training: 100%|██████████| 40/40 [00:33<00:00, 1.20it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.17it/s]
- Epoch 2: Validation loss = 1.8959488968054454, Validation accuracy: 0.5098905762036642
- Training: 100%|██████████| 40/40 [00:33<00:00, 1.20it/s]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.17it/s]
- Epoch 3: Validation loss = 1.9621925055980682, Validation accuracy: 0.5
- Training: 100%|██████████| 40/40 [01:18<00:00, 1.96s/it]
- Valid: 100%|██████████| 12/12 [00:09<00:00, 1.23it/s]
- Epoch 4: Validation loss = 1.4488181670506795, Validation accuracy: 0.508959099650383
- Training: 100%|██████████| 40/40 [02:03<00:00, 3.08s/it]
- Valid: 100%|██████████| 12/12 [00:09<00:00, 1.29it/s]
- Epoch 5: Validation loss = 1.660115271806717, Validation accuracy: 0.49278322358926135
- Training: 100%|██████████| 40/40 [02:00<00:00, 3.01s/it]
- Valid: 100%|██████████| 12/12 [00:09<00:00, 1.21it/s]
- Epoch 6: Validation loss = 2.0973944067955017, Validation accuracy: 0.5
- Training: 100%|██████████| 40/40 [01:33<00:00, 2.33s/it]
- Valid: 100%|██████████| 12/12 [00:03<00:00, 3.13it/s]
- Epoch 7: Validation loss = 0.7050559955338637, Validation accuracy: 0.747499868273735
- Validation accuracy increased (0.7065854867299398 --> 0.747499868273735). Model saved
- Training: 100%|██████████| 40/40 [01:23<00:00, 2.09s/it]
- Valid: 100%|██████████| 12/12 [00:09<00:00, 1.23it/s]
- **Epoch 8: Validation loss = 0.4437953482071559, Validation accuracy: 0.8310285409291586**
- Validation accuracy increased (0.747499868273735 --> 0.8310285409291586). Model saved
- Training: 100%|██████████| 40/40 [02:03<00:00, 3.09s/it]
- Valid: 92%|█████████ | 11/12 [00:08<00:00, 1.17it/s]
- Epoch 9: Validation loss = 1.2620619237422943, Validation accuracy: 0.6665227264165878
- Valid: 100%|██████████| 12/12 [00:09<00:00, 1.27it/s]

- Process finished with exit code 0