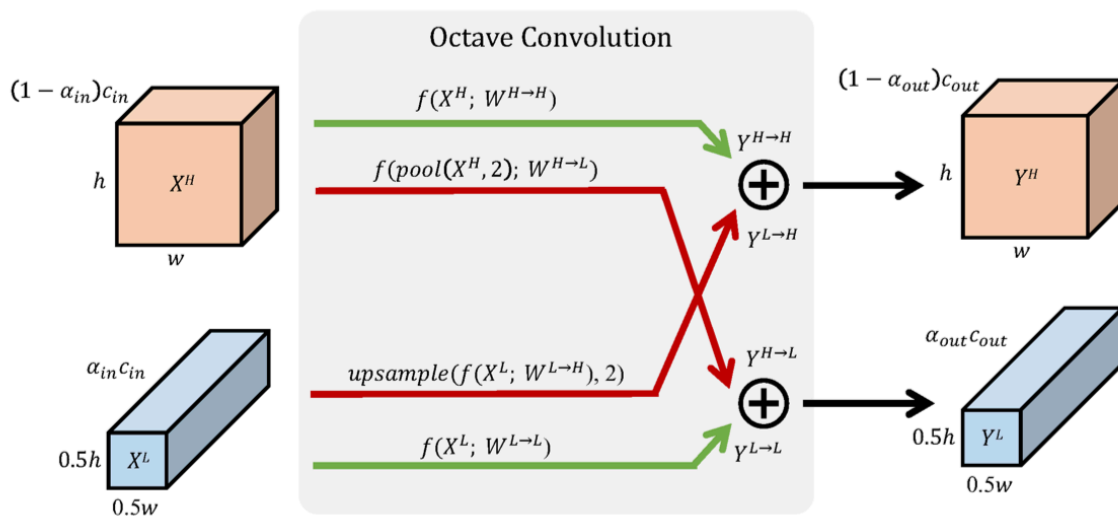
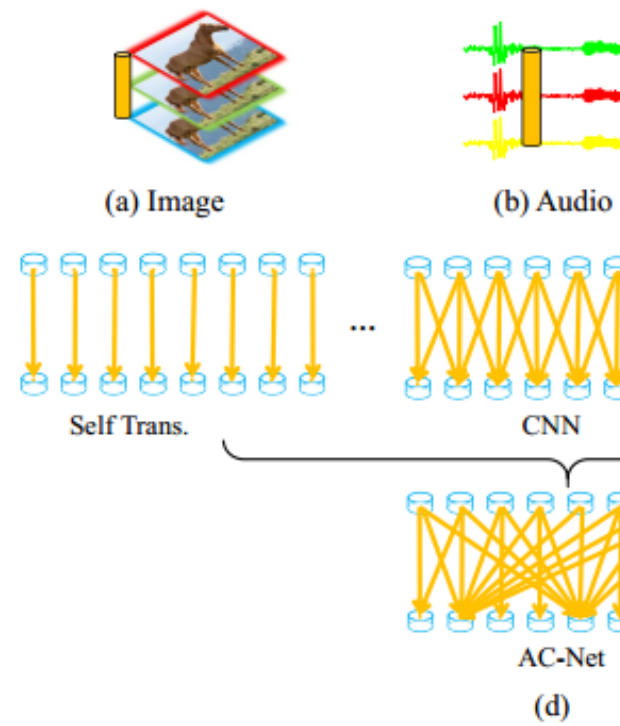

Beyond Convolution

OctaveConv_pytorch

Pytorch implementation of recent operators

This is **third parity** implementation(un-official) of Following Paper. 1. Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution(ICCV 2019). paper





2. Adaptively Connected Neural Networks.(CVPR 2019) paper

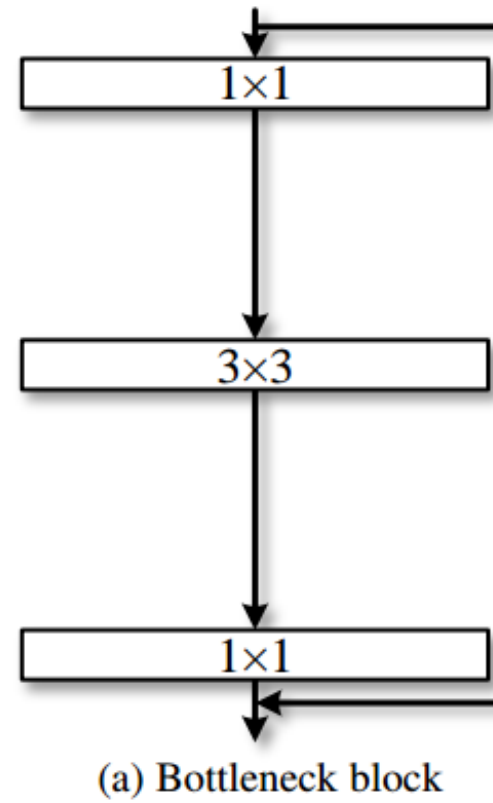


Fig. 2: Comparison between
posed Res2Net module (the

3. Res2net:A New Multi-scale Backbone Architecture(PAMI 2019) paper

4. ScaleNet:Data-Driven Neuron Allocation for Scale Aggregation Networks (CVPR2019) paper

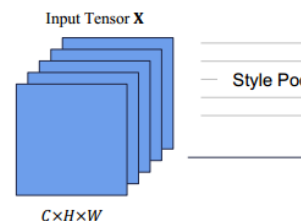
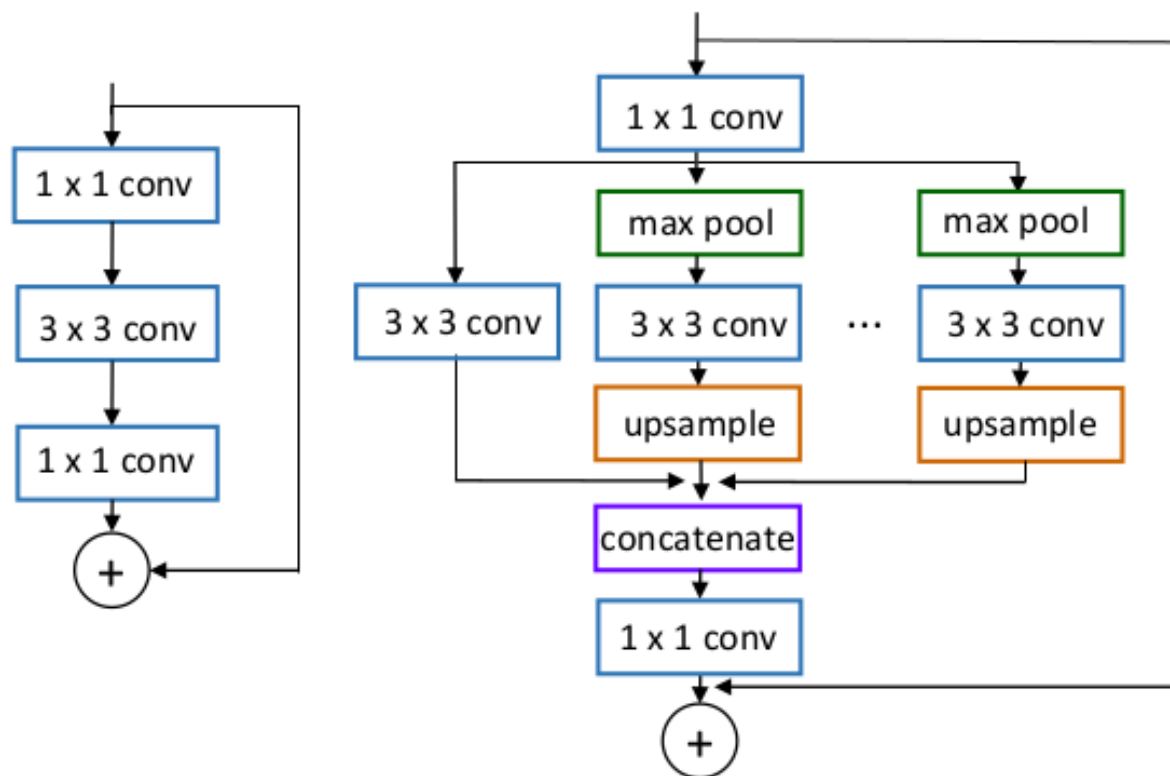


Figure 1: A Style-based Recalibration Module for Convolutional Neural Networks

5. SRM : A Style-based Recalibration Module for Convolutional Neural Networks paper
6. SEnet: Squeeze-and-Excitation Networks(CVPR 2018) paper
7. GEnet: Exploiting Feature Context in Convolutional Neural Networks(NIPS 2018) paper
8. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks paper
9. SK-Net: Selective Kernel Networks(CVPR 2019) paper
10. More Net will be added.

Plan

1. add Res2Net bolock with SE-layer (done)
2. add Adaptive-Convolution: both pixel-aware and dataset-aware (done)
3. Train code on Imagenet. (done)
4. Add SE-like models. (done)

-
5. Keep tracking with new proposed operators. (-)

Usage

check model files under the fig/nn floder.

```
1 from lib.nn.OctaveResnet import resnet50
2 from lib.nn.res2net import se_resnet50
3 from lib.nn.AdaptiveConvResnet import PixelAwareResnet50,
  DataSetAwareResnet50
4
5 model = resnet50().cuda()
6 model = se_resnet50().cuda()
7 model = PixelAwareResnet50().cuda()
8 model = DataSetAwareResnet50().cuda()
```

Training

see exp floder for the detailed information

Checkpoint

Reference and Citation:

1. OctaveConv: MXNet implementation here
2. AdaptiveCov: Offical tensorflow implementation here
3. ScaleNet: here
4. SGENet:here

Please consider cite the author's paper when using the code for your research. ## License MIT License