“Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network”

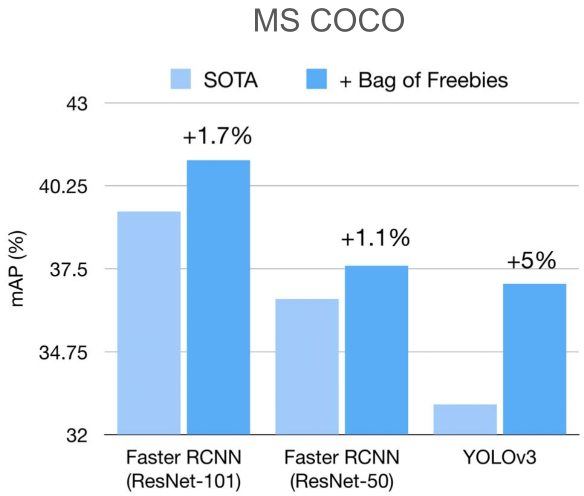
Compiled by Michael M. Pieler

History

- “Bag of tricks/freebies”[1,2]

ImageNet (ILSVRC2012)

Model	FLOPs	top-1	top-5
ResNet-50 [9]	3.9 G	75.3	92.2
ResNeXt-50 [27]	4.2 G	77.8	-
SE-ResNet-50 [12]	3.9 G	76.71	93.38
SE-ResNeXt-50 [12]	4.3 G	78.90	94.51
DenseNet-201 [13]	4.3 G	77.42	93.66
ResNet-50 + tricks (ours)	4.3 G	**79.29**	**94.63**



- In the meantime new powerful architectures and tricks!

[1] Bag of Tricks for Image Classification with Convolutional Neural Networks
[2] Bag of Freebies for Training Object Detection Neural Networks

Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network[3]

- Studied several CNN-related techniques and how they can be assembled into a single network.

- Goal: Improve
  - Accuracy
  - Robustness (mean corruption error “mCE”)
  - Throughput (images/sec, instead of FLOPS, because they are not proportional to the inference speed.)

[3] Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network & https://github.com/clovaai/assembled-cnn

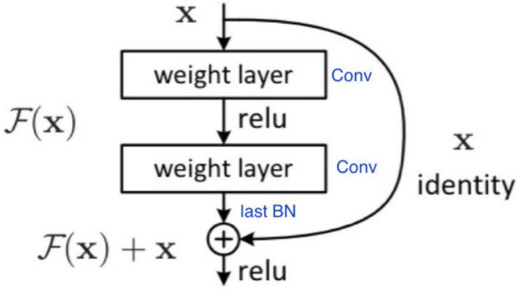
Focus on network tweaks & regularization

- "Network tweaks are methods that modify the CNN architectures to be more efficient"

- "Regularization is a method that prevents overfitting by increasing the training data through data augmentation processes… or by limiting the complexity of the CNN"

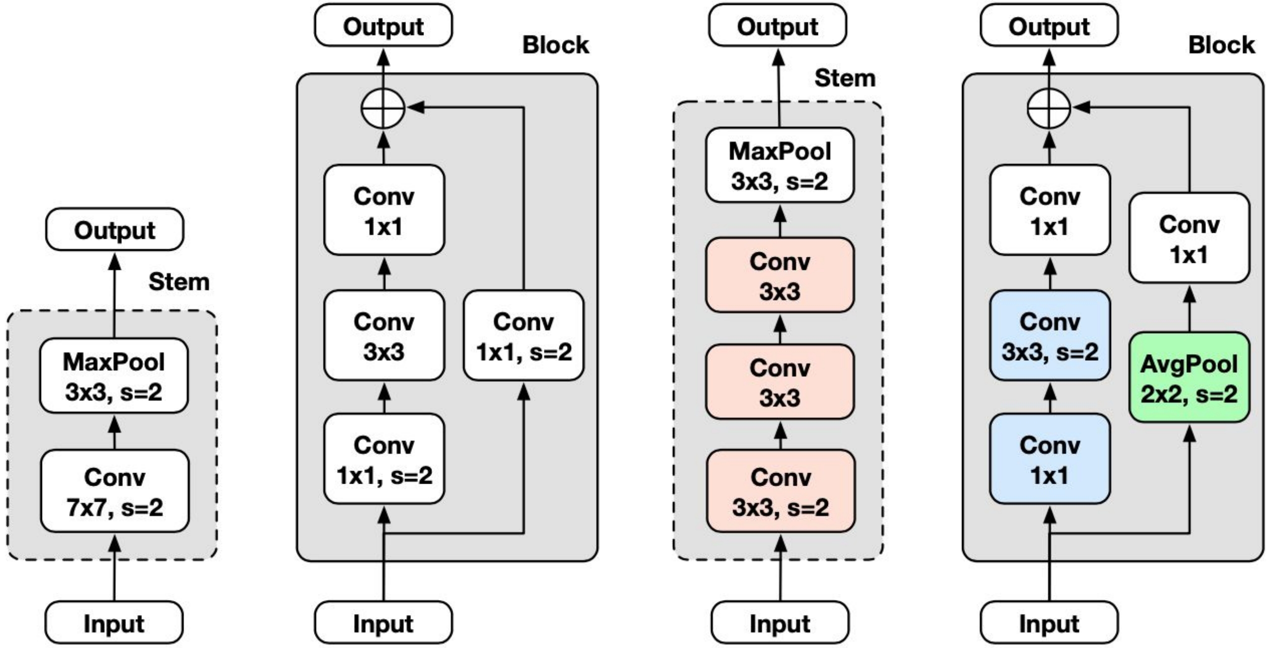
Training Procedure

- Image preprocessing: normalize & random crop, resize, flip
- Hyperparameter: initial lr = 0.4, wd = 0.0001, epochs = 120, optimizer: SGD, mom = 0.9
- Learning rate warmup: 0 to initial lr in the first 5 epochs
- Cosine learning rate decay
- Mixed precision training (FP32 & FP16)
- **Zero γ init:** Initialize γ = 0 for all BN layers at the end of a res-block. Returns identity at the beginning!



$$\gamma\left(\frac{(x-\mu_B)}{\sigma_B+\epsilon}\right)+\beta$$

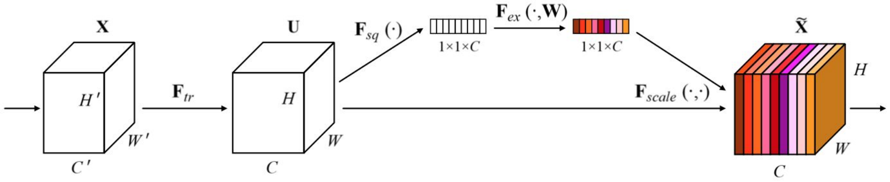
Network tweaks: ResNet-D



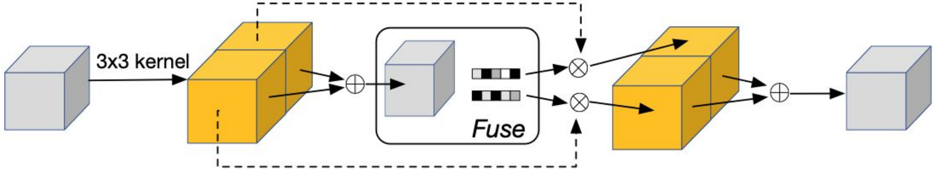
(a) ResNet (b) ResNet-D

Network tweaks: Channel attention

- **Squeeze & Excite (SE)**[4]

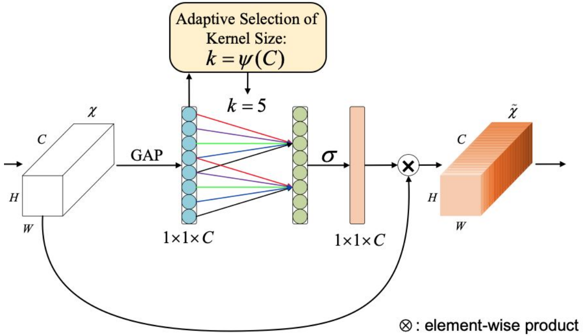


- **Selective Kernel (SK)**[5]



One *3x3 kernel*
with doubled channel output

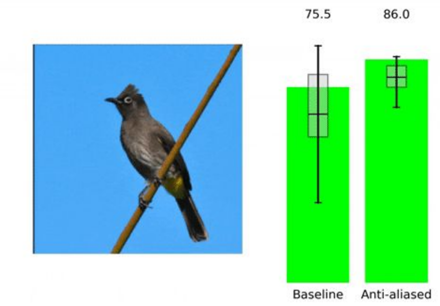
- ***Outlook: Efficient Channel (EC)***[6]



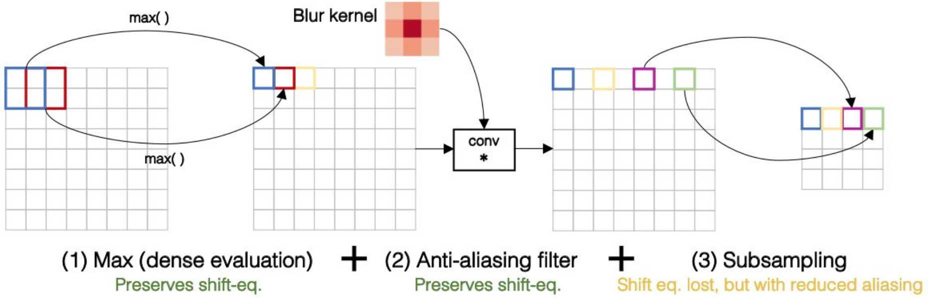
[4] Squeeze-and-Excitation Networks
[5] Selective Kernel Networks
[6] ECA-Net: Efficient Channel Attention for D-CNNs

Network tweaks: AA downsampling & Big Little Net

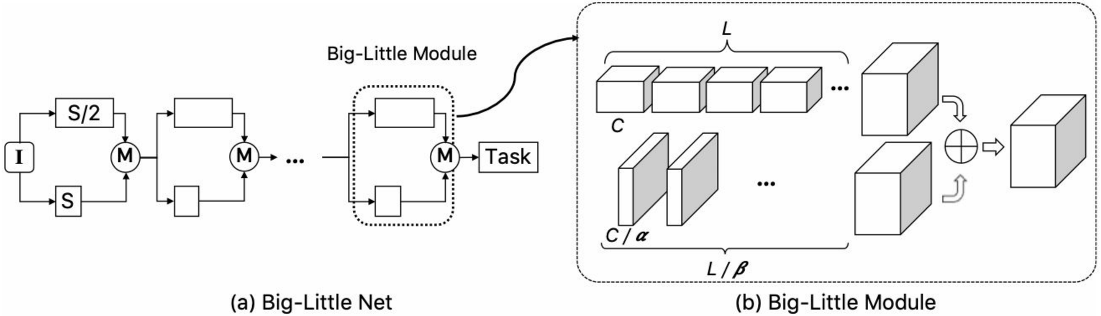
- **Anti-alias downsampling[7]:** Improves the shift-equivariance



Anti-aliased
(MaxBlurPool)



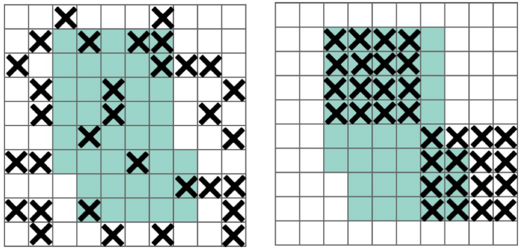
- **Big Little Net[8]**



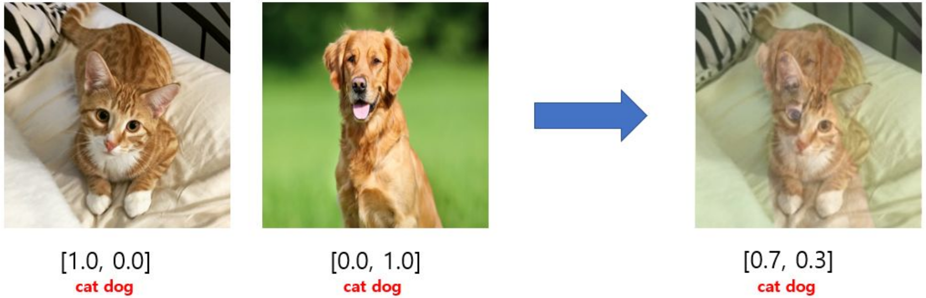
[7] [Making Convolutional Networks Shift-Invariant Again](#)
& https://github.com/adobe/antialiased-cnns
[8] [Big Little Net](#)

Regularization

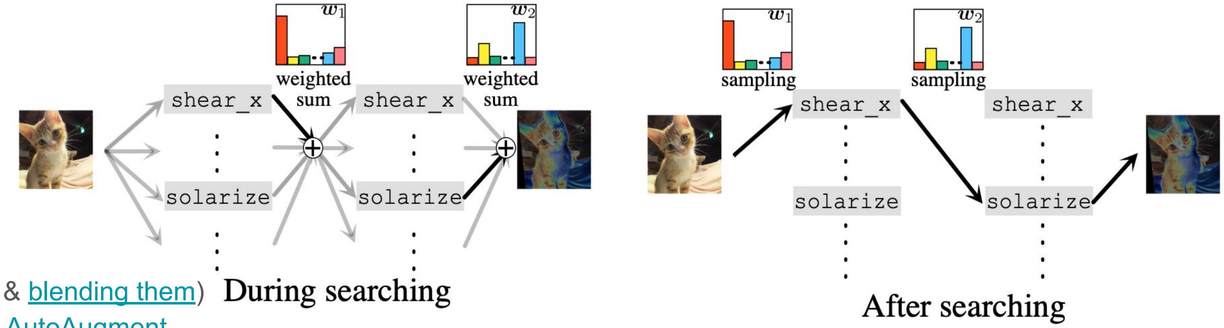
- **DropOut and DropBlock**[9]



- **MixUp**[10]



- **AutoAugment**[11]



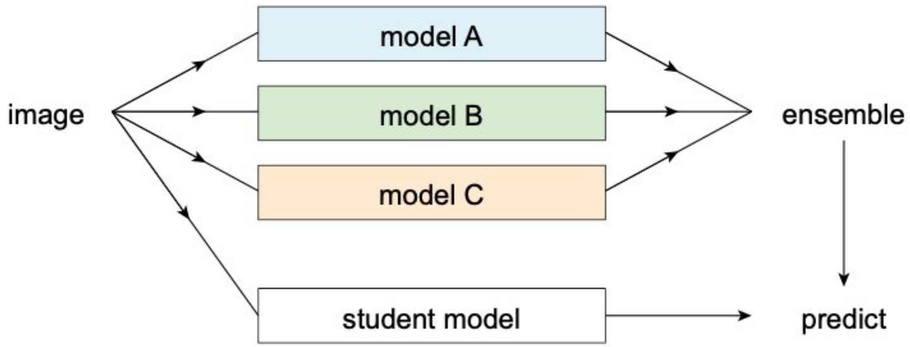
[9] DropBlock
[10] MixUp & blog post (CutOut, Ricap, CutMix & blending them)
[11] AutoAugment, Fast AutoAugment & Faster AutoAugment

Regularization

- **Label Smoothing**[12]
  - avoid overconfidence
  - smoothing factor f = 0.1
  - Pos. p_i: 1 → 1 - f = 0.9
  - Neg. p_j: 0 → f / (N-1)
  - different implementations!

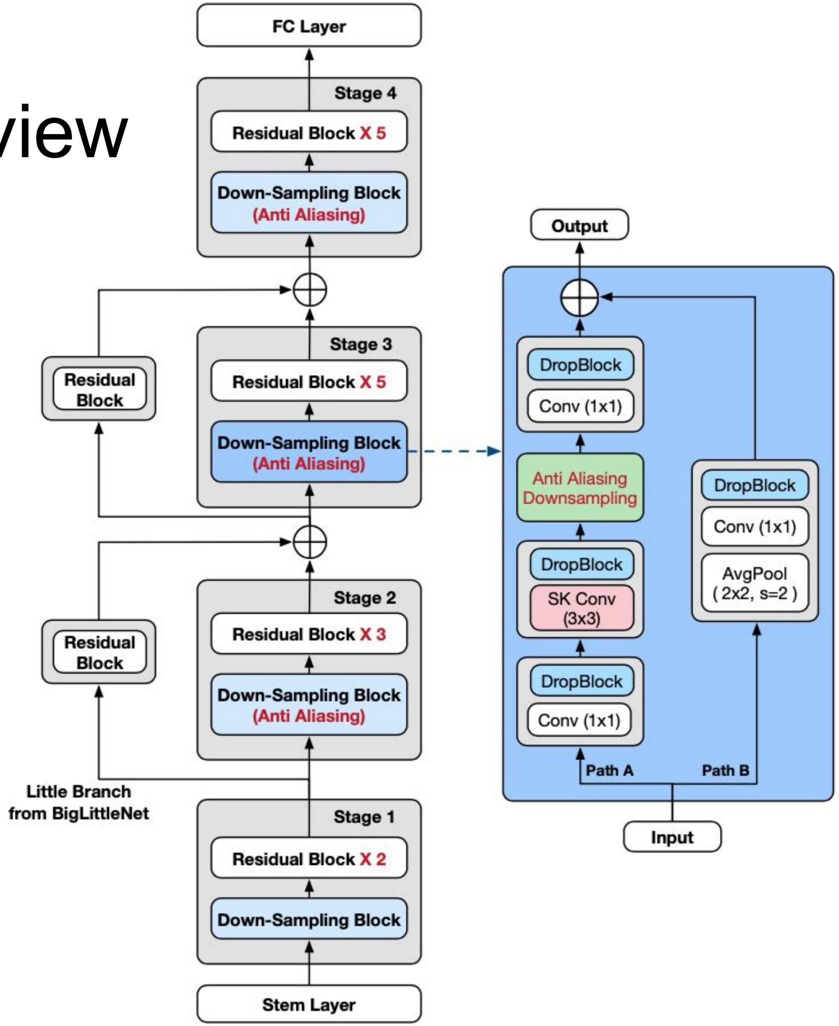
$$(1 - eps)\,(-\log(p_i)) + \sum_{j \neq i} \frac{eps}{N-1}(-\log(p_j))$$

- **Knowledge Distillation**[13]



[12] Regularizing Neural Networks by Penalizing Confident Output Distributions, When Does Label Smoothing Help? & blog post
[13] Distilling the Knowledge in a Neural Network & Data Distillation

Network overview



Results

ImageNet (ILSVRC2012)

Model	top-1	mCE	throughput
EfficientNet B4 [34]+AutoAugment [4]	83.0	60.7	95
EfficientNet B6 [34]+AutoAugment [4]	84.2	60.6	28
EfficientNet B7 [34]+AutoAugment [4]	84.5	59.4	16
ResNet-50 [9] (baseline)	76.3	76.0	536
Assemble-ResNet-50 (ours)	82.8	48.9	312
Assemble-ResNet-152 (ours)	84.2	43.3	143

Questions?