COMPUTER VISION

CNN ARCHITECTURES

AGENDA

This video lecture focuses on understanding:

- In depth explanation on shallow networks: LeNet, AlexNet.
- In depth explanation on initial deep networks: ZFNet, VGGNet.
- In depth explanation on advanced networks: GoogleNet, ResNet and its variants.
- Brief explanation on state-of-the art networks: MobileNet, Efficient Net and its variants.

INDEX

1. LeNet-5 (1998)

- **Task**: Handwritten digit classification by Yann LeCun
- **Dataset**: MNIST data
  - Input: a small single channel image
  - Output: 10 outputs corresponding to the 10 digits 0-9
  - Size: 60,000 training images,10,000 test images

1. LeNet-5 (1998)

- **Architecture**: 3 Conv + 2 Avg pooling + 1 FC layer + 1 FC-softmax layer
- **Applications**: used by banks to recognize handwritten numbers on digitized checks
- **Advantages**: tolerant of various transformations like rotation and scaling

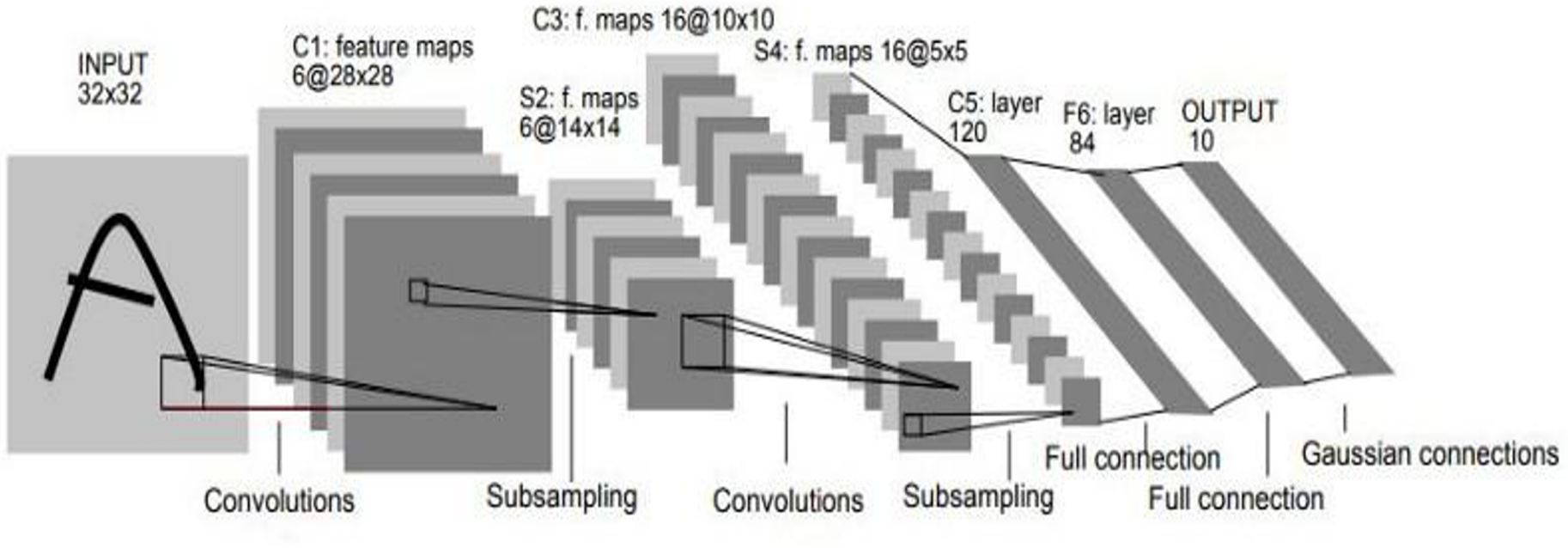


Image Source: Gradient Based Learning Applied to Document Recognition, LeCun et al. (1998)

1. LeNet-5 (1998)

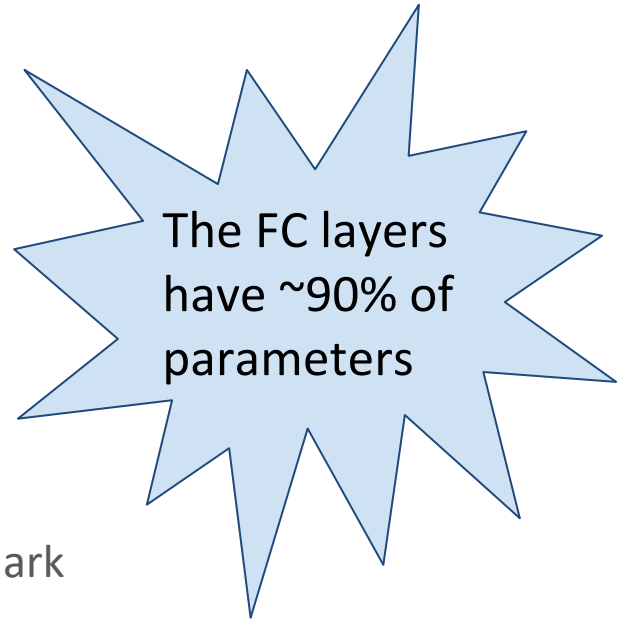
Layer	Activation Function	No. of filters and Filter Size	Feature Map	No. of trainable parameters
Conv-1	tanh	6 : (5 × 5)	28 × 28 × 6	156
Avg. Pool-1	-	2 × 2	14 × 14 × 6	12
Conv-2	tanh	16 : (5 × 5)	10 × 10 × 16	1516
Avg. Pool-2	-	2 × 2	5 × 5 × 16	32
Conv-3	tanh	120 : (5 × 5)	1 × 1 × 120	48120
FC-1	tanh	-	84	10164
FC-2 Output	softmax	-	10	840

2. AlexNet (2012)

Task: ImageNet classification

Dataset: Input: RGB images

- Output: 1000 outputs corresponding to the 1000 classes
- Size: 1.2M training images, 100K test images

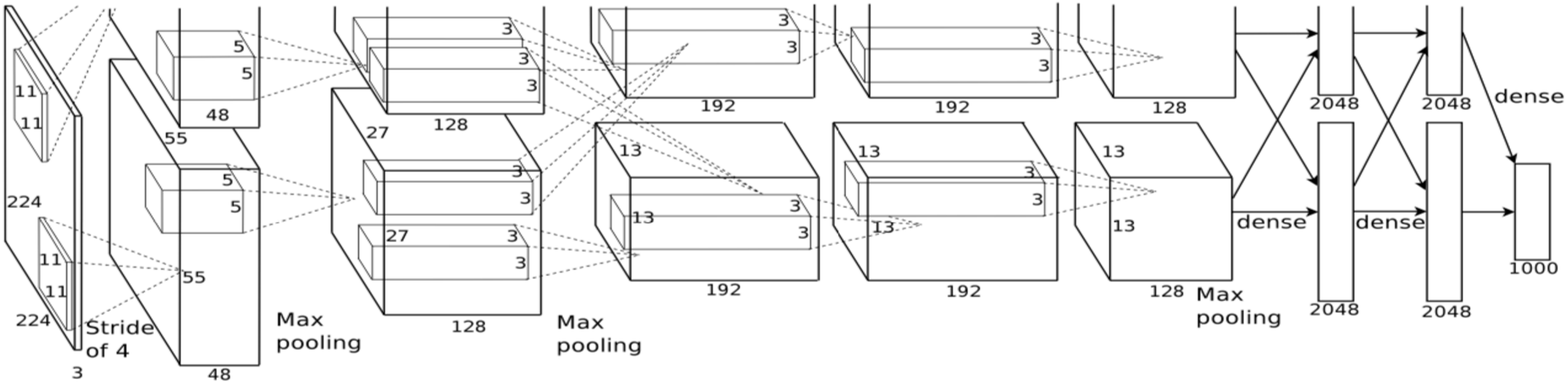


Key highlights:

- First CNN to be able to classify imagenet images successfully and improved benchmark performance (top-5) from 26% to 15%
- Introduced ReLU nonlinearity
- Used data Augmentation, dropout, L2 weight decay 5e-4
- Learning rate 1e-2. Reduced by 10 manually when val accuracy plateaus
- 7 CNN ensemble: 18.2% improved to 15.4%
- Used two GPUs for training

2. AlexNet (2012)

Architecture: 5 Conv + 3 Max Pool + 2 FC layers + 1 FC-softmax layer



Source: ImageNet Classification with Deep Convolutional Neural Networks, Krizhevsky et al. (2012)

2. AlexNet (2012)

Layer	Activation Function	No. of filters and Filter Size	Stride and Padding	Feature Map	No. of trainable parameters
Input	-	-		227x227x3	-
Conv-1	ReLU	96 : (11 × 11)	4 stride & 0 padding	55 × 55 × 96	34944
Max. Pool-1	-	3 × 3	2 stride	27 × 27 × 96	-
Conv-2	ReLU	256 : (5 × 5)	1 stride & 2 padding	27 × 27 × 256	614,656
Max. Pool-2	-	3 × 3	2 stride	13 × 13 × 256	-
Conv-3	ReLU	384 : (3 × 3)	1 stride & 1 padding	13 x 13 x 384	885,120
Conv-4	ReLU	384 : (3 × 3)	1 stride & 1 padding	13 x 13 x 384	1,327,488
Conv-5	ReLU	256 : (3 × 3)	1 stride & 1 padding	27 x 27 x 256	884,992
Max. Pool-3		3 × 3	2 stride	6 x 6 x 256	-
FC-1, Fc-2, Fc3-	ReLU Softmax	4096, 1000	-	4096 × 1, 1000 * 1	37,752,832 + 16,781,312 + 4,097,000

Total Parameters: ~62M

3. ZF Net (2013)

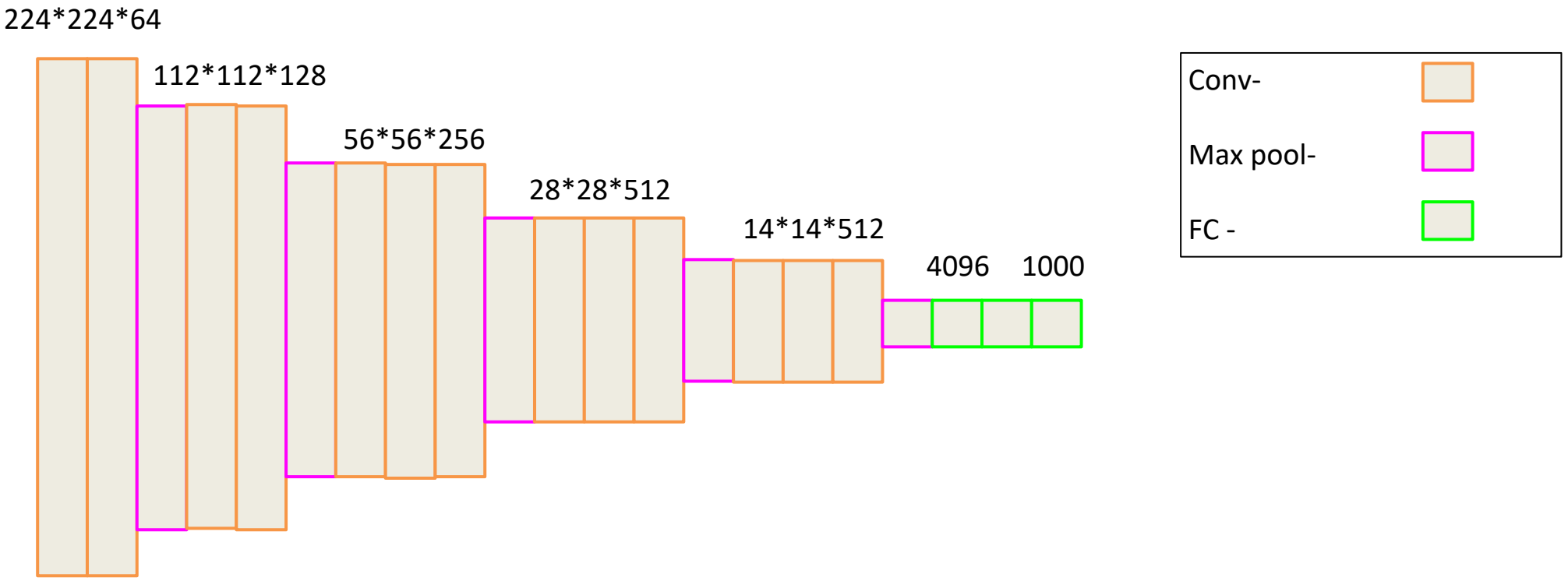
- **Task**: ImageNet Classification winner in 2013
- **Dataset**: ImageNet Data
- **Key highlights**:
  - Similar to AlexNet
  - Used (7 * 7) convolutions and large number of filters (512, 1024, 512 instead of 384, 384, 256 respectively)
  - Improved benchmark performance (top-5) from 15.4% to 14.8%. Later they bought it down to 11.2%
  - Access to better hardware
  - **Architecture**: 5 Conv + 3 Max Pool + 2 FC layers + 1 FC-softmax layer

4. VGG Net (2014)

- **Task**: ImageNet classification (runner in 2014 ILSVRC)
- **Dataset**: ImageNet data
- **Key highlights**:
  - It has two common variants: VGG-16 and VGG-19
  - Known for its simplicity: 3*3 filters
  - It was very slow to train so they used pretraining (first trained smaller versions of VGG with less weight layers and then used them as initializations for the larger, deeper networks)
  - L2 regularization is used, and the weight decay is 5e-4
  - Dropout regularization for the first two fully-connected layers (dropout ratio set to 0.5)

4. VGG Net (2014)

Architecture: 16 layer deep -> 13 Conv + 5 Max pooling + 3 fully connected layer

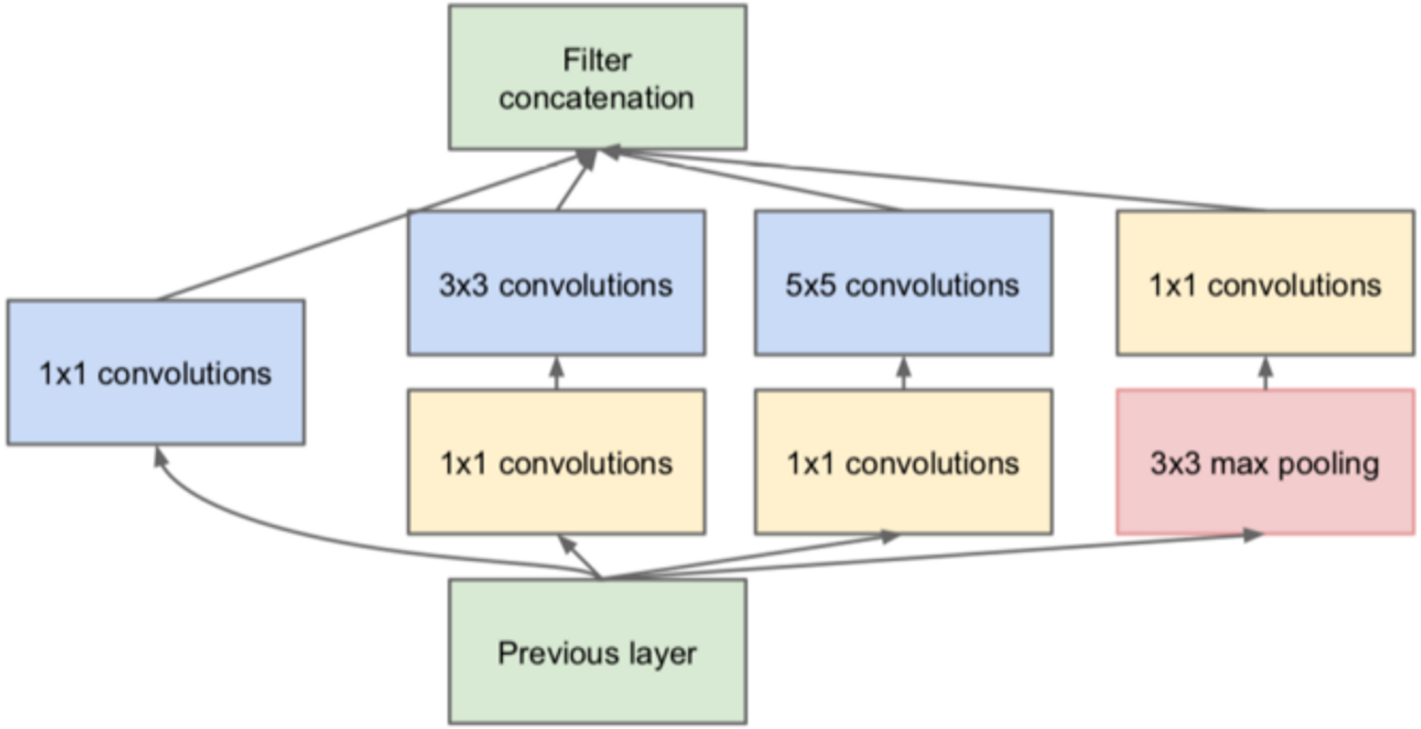


5. Google Net (2014)

- **Task**: ImageNet Classification (ILSVRC Winner in 2014)
- **Key Highlights (top-5 error 6.67%)**:
  - **Idea**:- started with an assumption that most of the activations in a deep network are redundant because of correlations between them.
  - Introduced inception module to build sparse CNN
  - It has 5*5 filters which capture global features and 3*3 filters which capture distributed features
  - It has bottleneck layer(1X1 convolutions) to reduce input channel depth
  - It replaced FC layers with global average pooling layer
  - It has two auxiliary outputs to improve convergence

5. Google Net (2014)

Architecture: 2 CONV + 9 Inception modules + 4 Max pool + 1 avg pool + dropout layer + FC softmax

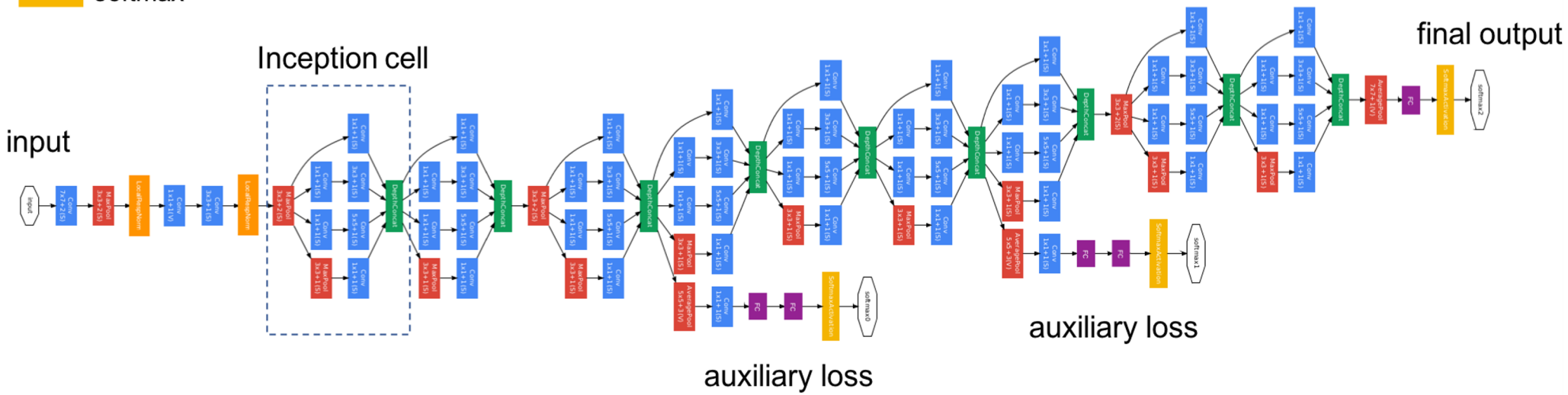


Source:Going deeper with convolutions

5. Google Net (2014)

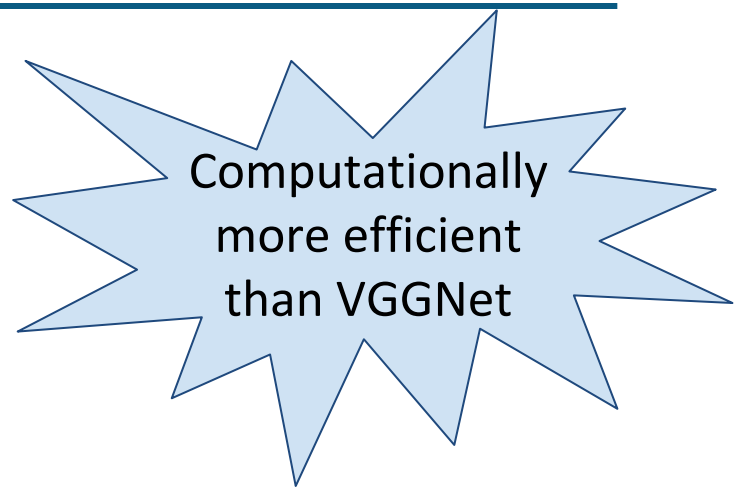
Architecture:

- convolution
- max pooling
- channel concatenation
- channel-wise normalization
- fully-connected layer
- softmax



Source:Going deeper with convolutions

6. ResNet (2015)



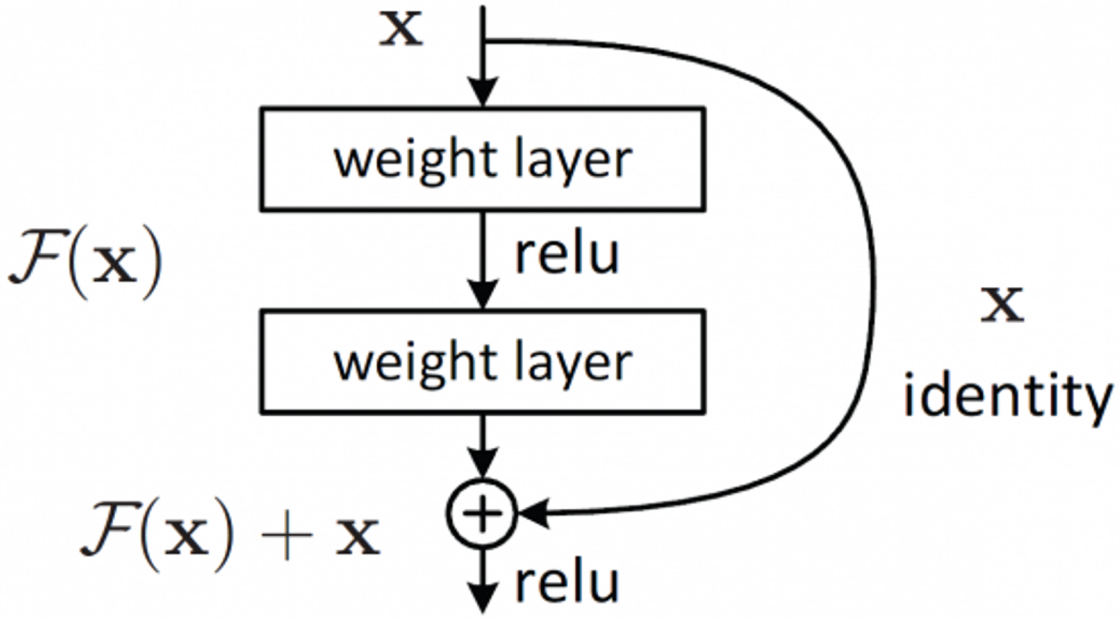
Key highlights (top-5 error to 3.6%):

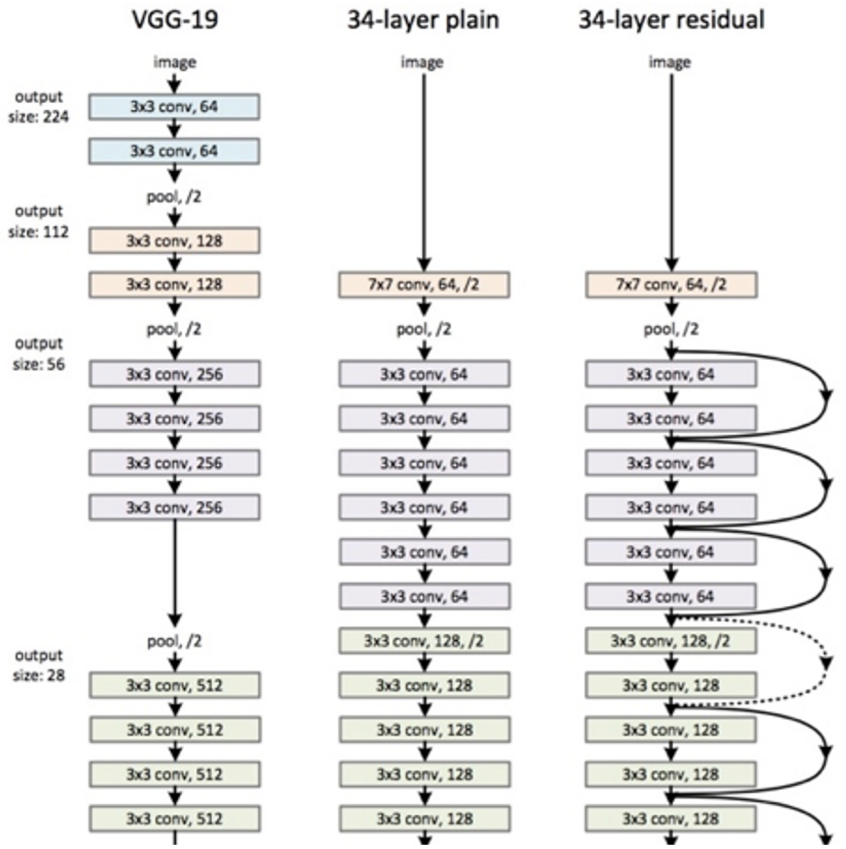
- First model to use ultra deep network.
- Achieved 1st place in all five main tracks (Imagenet classification, detection, localization and coco detection, coco segmentation).
- Introduced residual connection to deal with vanishing gradient issues.
- It has a global average pooling followed by the classification layer.
- It used batch Normalization after every CONV layer.
- Can be considered an ensemble of multiple networks.

Training:- Used Xavier/2 initialization from He et al. It used SGD with 0.9 momentum, and LR of 0.1, divided by 10 when validation error plateaus.

6. ResNet (2015)

Architecture: ImageNet classification





Source: Deep Residual Learning for Image Recognition, Kaiming He (2015)

7. ILSVRC 2016 Winner

Key highlights:

- CUImage was the winner in 2016
- Classification error is down to 3.0% from 3.6% last year.
- Used ensemble of different pretrained models.
- Analyzed the wrongly predicted images in detail and improved the network.

7. ILSVRC 2017 Winner

Squeeze & excitation network (SENet):

- Won in 2017 with 2.251% top-5 error on the test set.
- Introduced squeeze and Excitation block that can be added to a Conv Layer.
- Added parameters to each channel of a convolutional block so that the network can adaptively adjust the weighting of each feature map.
- Final model is an ensemble of SENets.

8. MobileNet (2017)

Key highlights:

- Developed by Google
- Small model size, low-latency, low-power model
- Particularly useful for mobile and embedded vision applications (designed for edge devices)
- Used Depthwise Separable Convolution
- Batch Normalization (BN) and ReLU are applied after each convolution
- Can be used for classification, detection, and segmentation

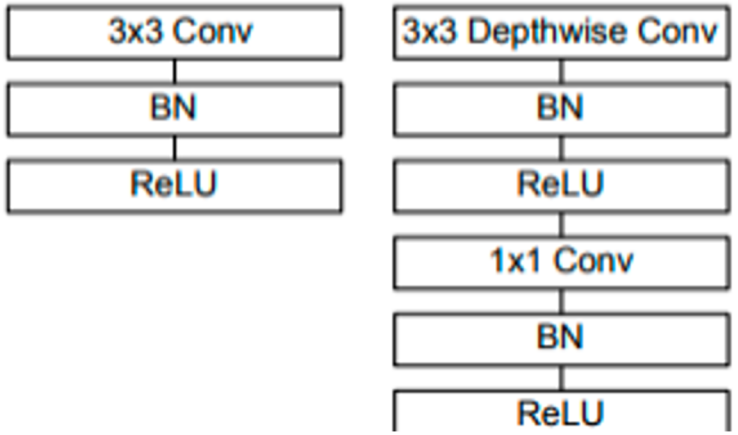
Width multiplier (α) and resolution multiplier (ρ) allow model developers to trade off latency or accuracy for speed and low size depending on their requirements.

8. MobileNet (2017)

Architecture: 28 layers-> 14 Conv + 13 DW Conv + Global Avg Pool + FC (Softmax)

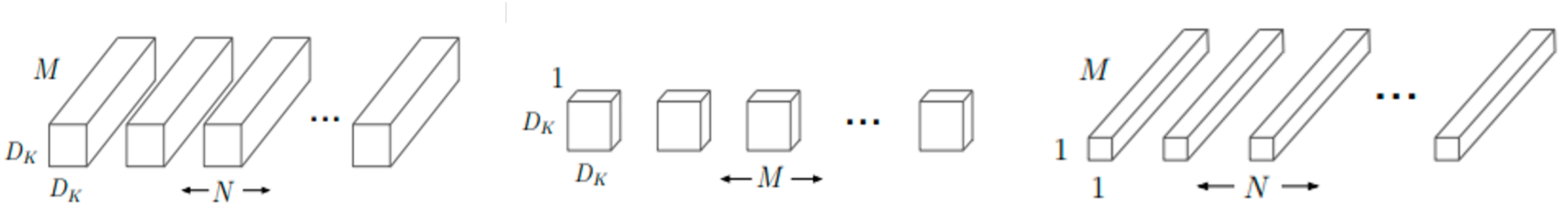
Left: Standard convolutional layer with batchnorm and ReLU.

Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.



***Source**: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*

8. MobileNet (2017)



a) Standard Convolution Filters

b) Depthwise Convolution Filters

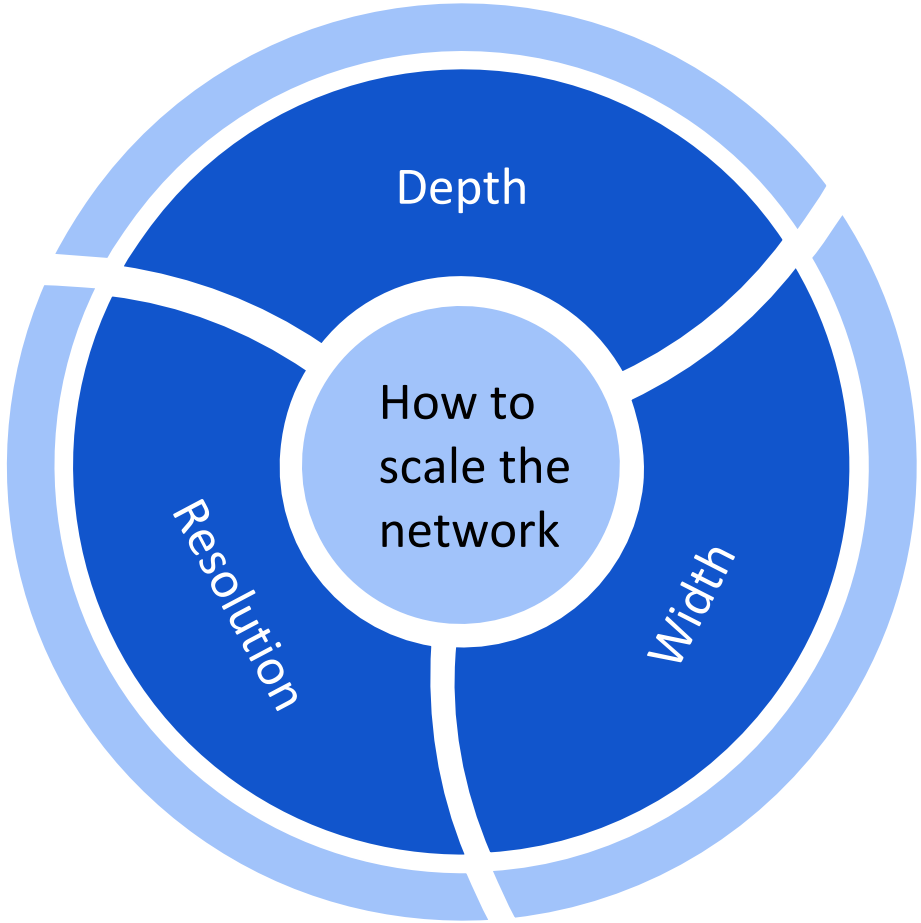
c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Fig: The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.

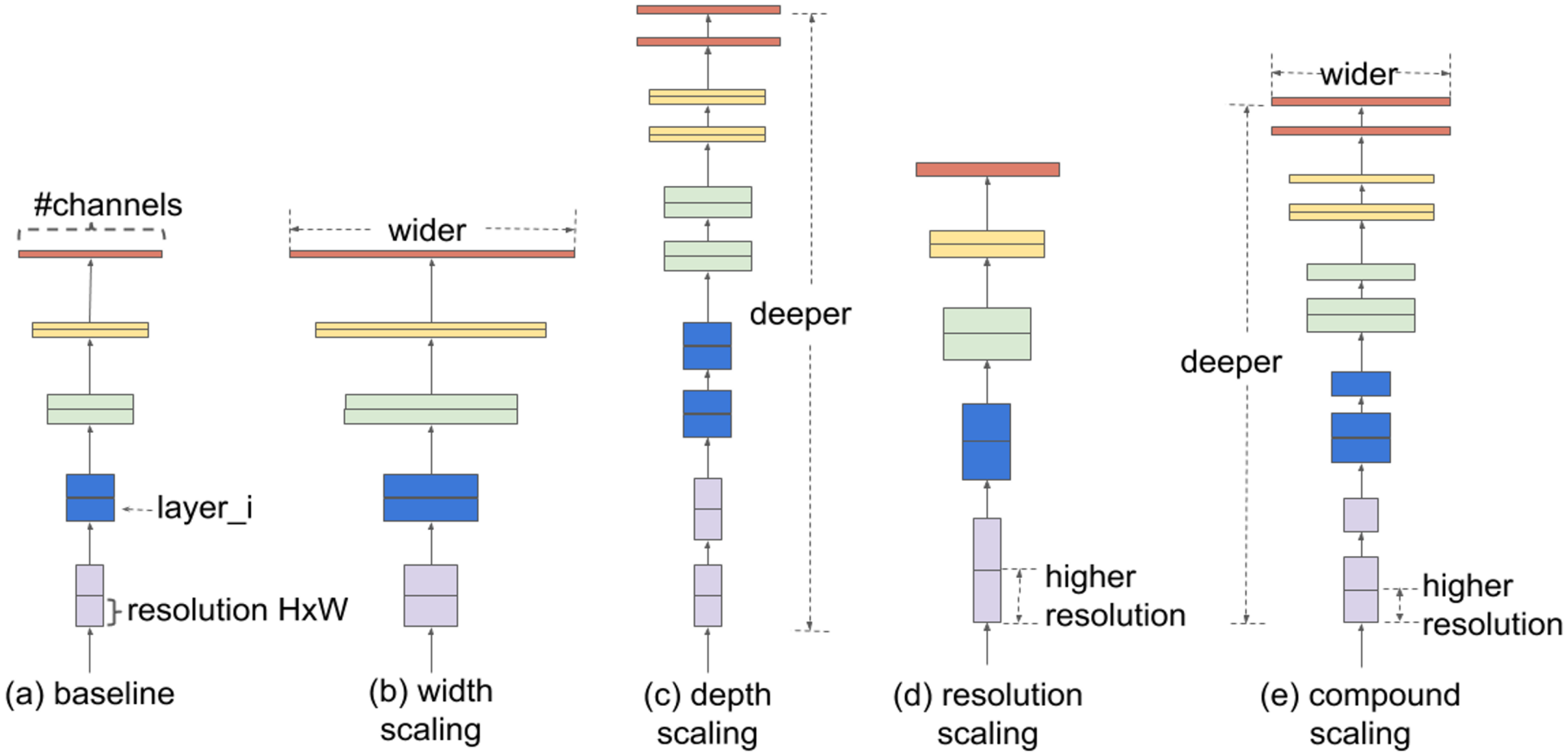
***Source**: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*

9. Efficient Net (2019)

- **Key highlights**:
  - Used compound coefficients to scale up CNNs
  - Idea:- instead of using arbitrary scaling, it uniformly scales each dimension with a fixed set of scaling coefficients.
  - Used grid search to find appropriate scaling coefficient under a fixed resource constraint.
  - Used EfficientNet-B0 as baseline network
  - Top-5 accuracy is 94.5%
  - EfficientNet-B7 has the state-of-the-art 97.1% top-5 accuracy on ImageNet data



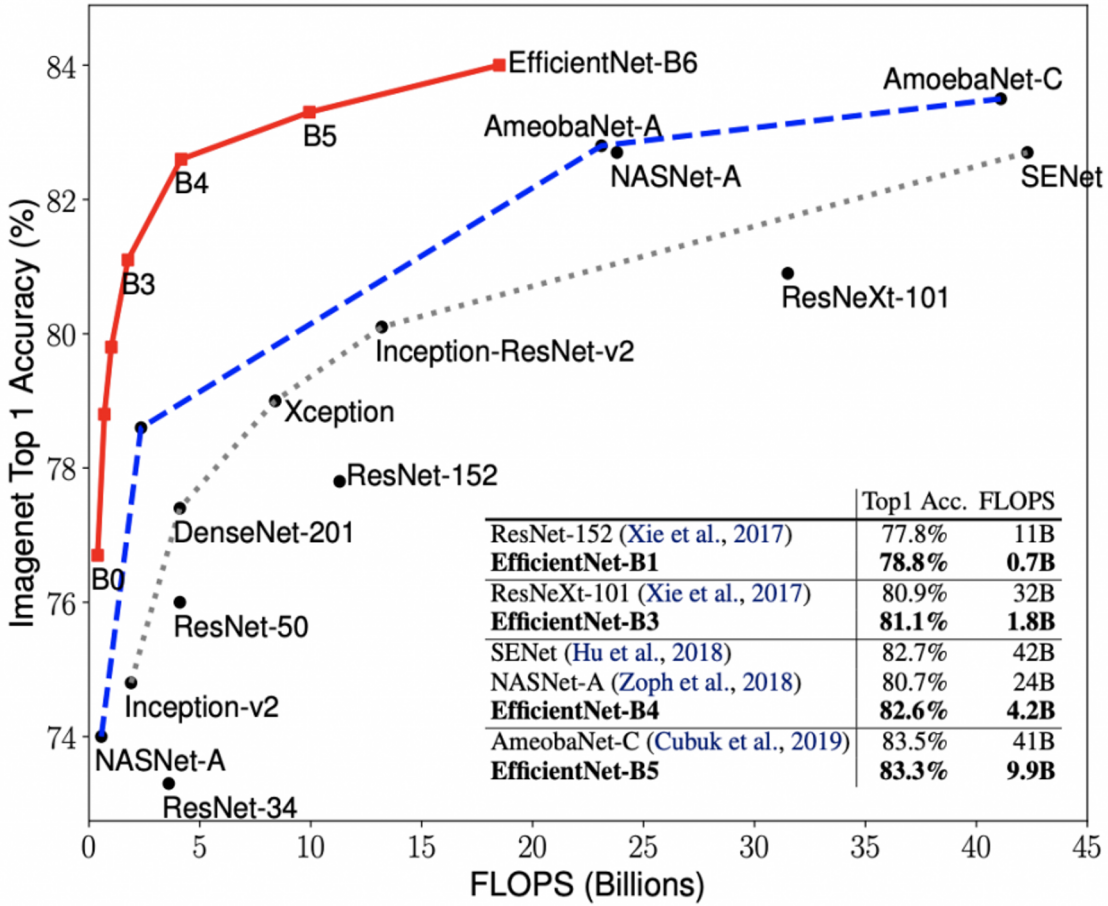
9. Efficient Net (2019)



Source: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

9. Efficient Net (2019)

- Model size vs. ImageNet Accuracy plot.
- EfficientNets significantly outperform other ConvNets.
- EfficientNet-B7 achieves the state-of-the-art 84.3% top-1 accuracy.
- EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152.

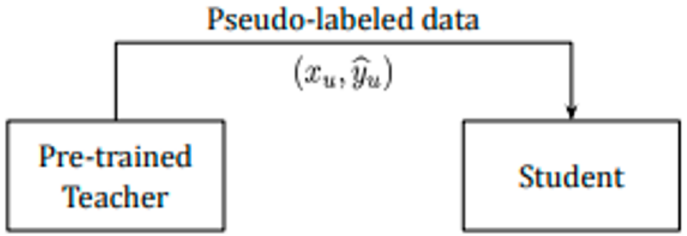


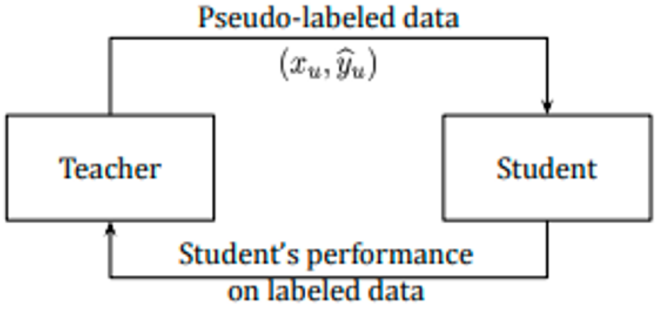
	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**78.8%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.1%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**82.6%**	**4.2B**
AmeobaNet-C (Cubuk et al., 2019)	83.5%	41B
EfficientNet-B5	**83.3%**	**9.9B**

Source: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

10. Meta Pseudo-Labels (2020)

- Semi-supervised learning method
- It has achieved state-of-the-art top-1 accuracy of 90.2% on ImageNet
- It uses teacher and student network pair.
- The teacher network is used to generate pseudo labels on unlabeled data. It is constantly adapted by the feedback of the student's performance on the labeled dataset.
- The student network is trained on the pseudo labeled images (learned by teacher network) and the labeled images.





Source: Meta Pseudo Labels

Summary

- We saw that most of the architectures stack multiple CONV, POOL, and FC layers
- There is a trend towards smaller filters and deeper architectures every passing year.
- There is trend towards reducing Fc layers.
- Typical architectures look like:

  [(CONV-RELU)*N-POOL]*M-(FC-RELU)*K-SOFTMAX

  where N is usually up to ~5, M is large, 0 <= K <= 2.

- Latest architectures such as ResNet/GoogLeNet/MobileNet challenge this paradigm and use advanced Conv layers.
- Trend towards small size and low-latency models.
- Trend towards semi supervised training methods.

Official Papers

- LeNet:- http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf
- AlexNet:- https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- ZF Net:- https://arxiv.org/abs/1311.2901
- VGGNet:- https://arxiv.org/abs/1409.1556
- GoogleNet:- https://arxiv.org/abs/1409.4842
- Resnet:- https://arxiv.org/abs/1512.03385
- Squeeze Net:- https://arxiv.org/abs/1709.01507
- MobileNets:- https://arxiv.org/pdf/1704.04861v1.pdf
- Efficient net:- https://arxiv.org/pdf/1905.11946.pdf
- Meta Pseudo Labels:- https://arxiv.org/abs/2003.10580