Deep Neural Networks

Machine Learning and Pattern Recognition

(Largely based on slides from Fei-Fei Li & Justin Johnson & Serena Yeung)

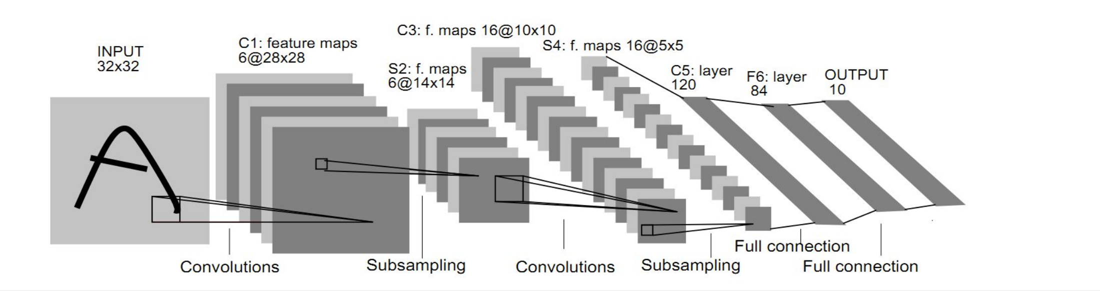
Prof. Sandra Avila

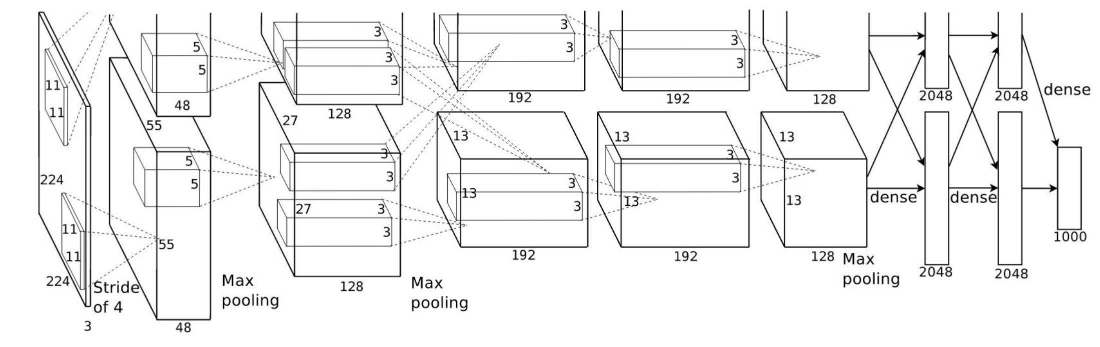
Institute of Computing (IC/Unicamp)

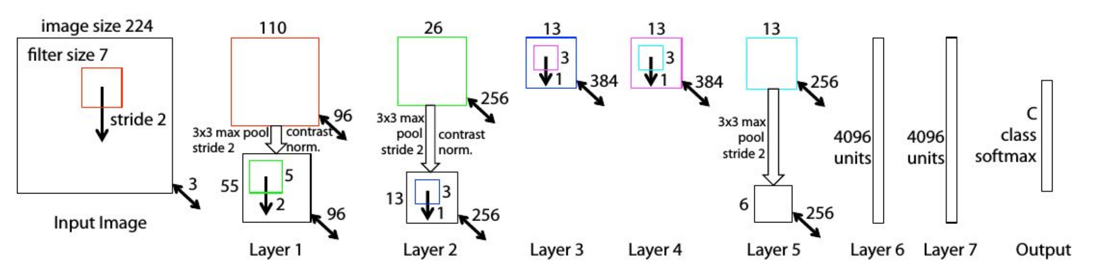
MC886/MO444, October 11, 2018

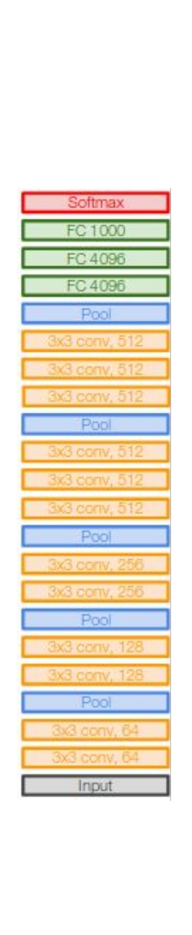
CNNs Architectures

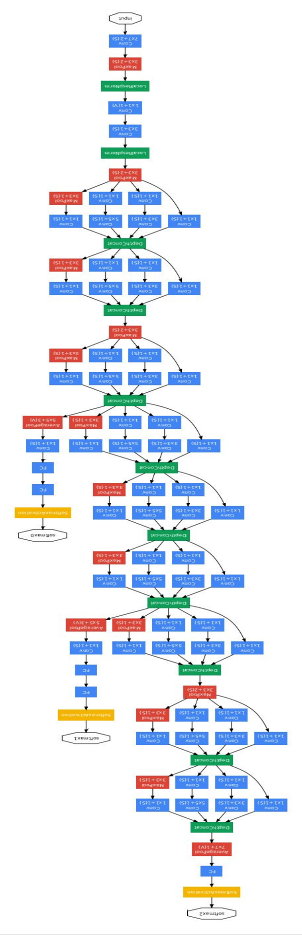
CNN-based Architectures

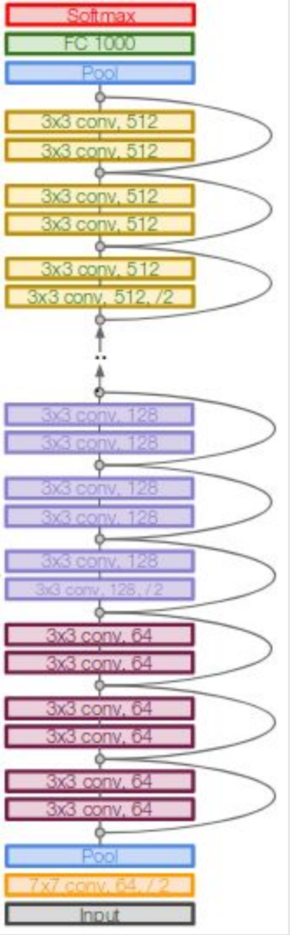












CNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

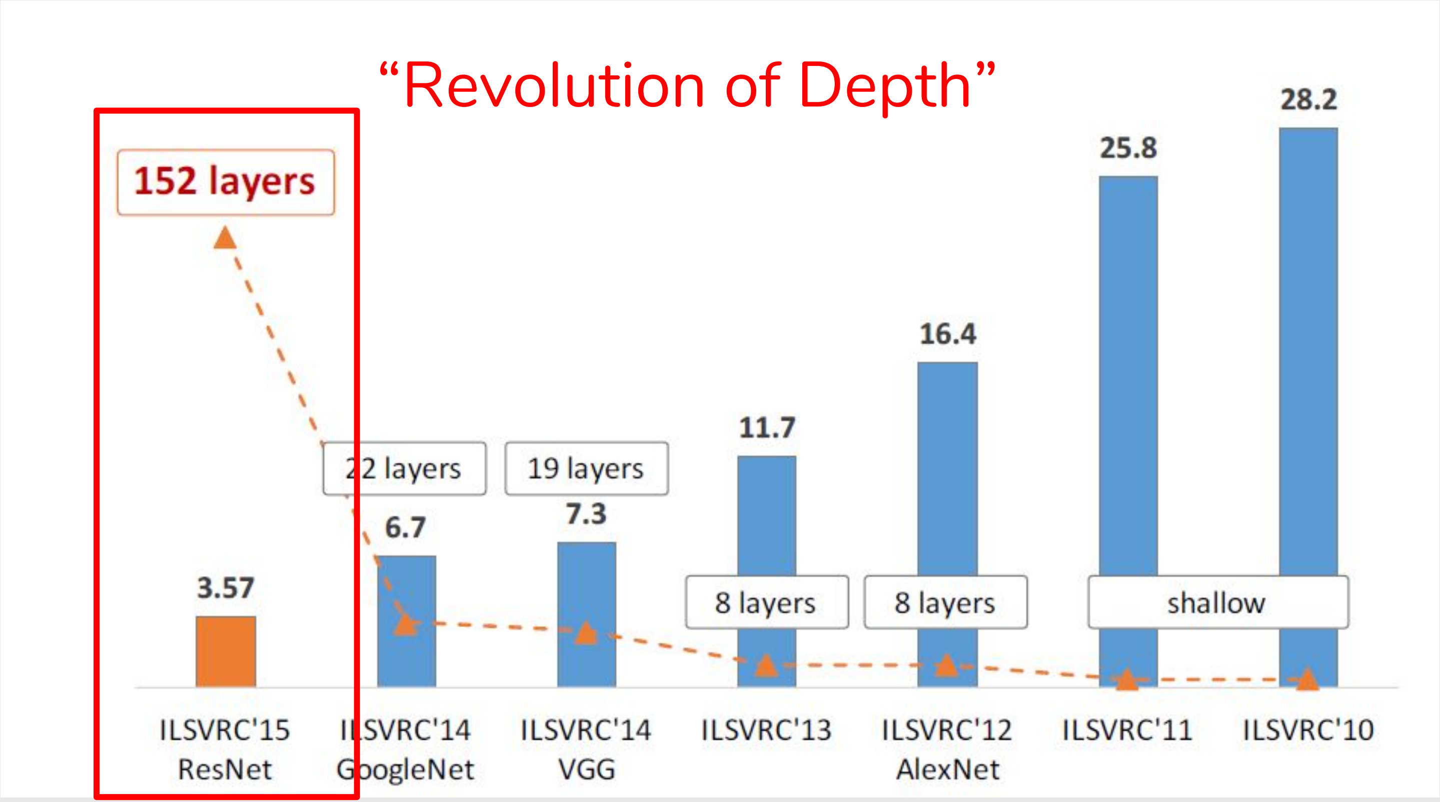
ResNet [He et al., 2015]

ResNet @ ILSVRC & COCO 2015 Competitions

1st place in ALL five main tracks

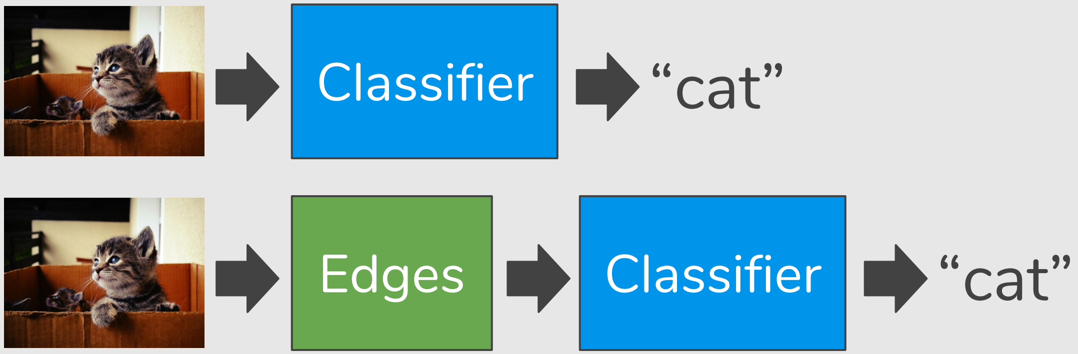
- ImageNet Classification: “Ultra-deep” 152-layer nets
- ImageNet Detection: 16% better than 2nd
- ImageNet Localization: 27% better than 2nd
- COCO Detection: 11% better than 2nd
- COCO Segmentation: 12% better than 2nd

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

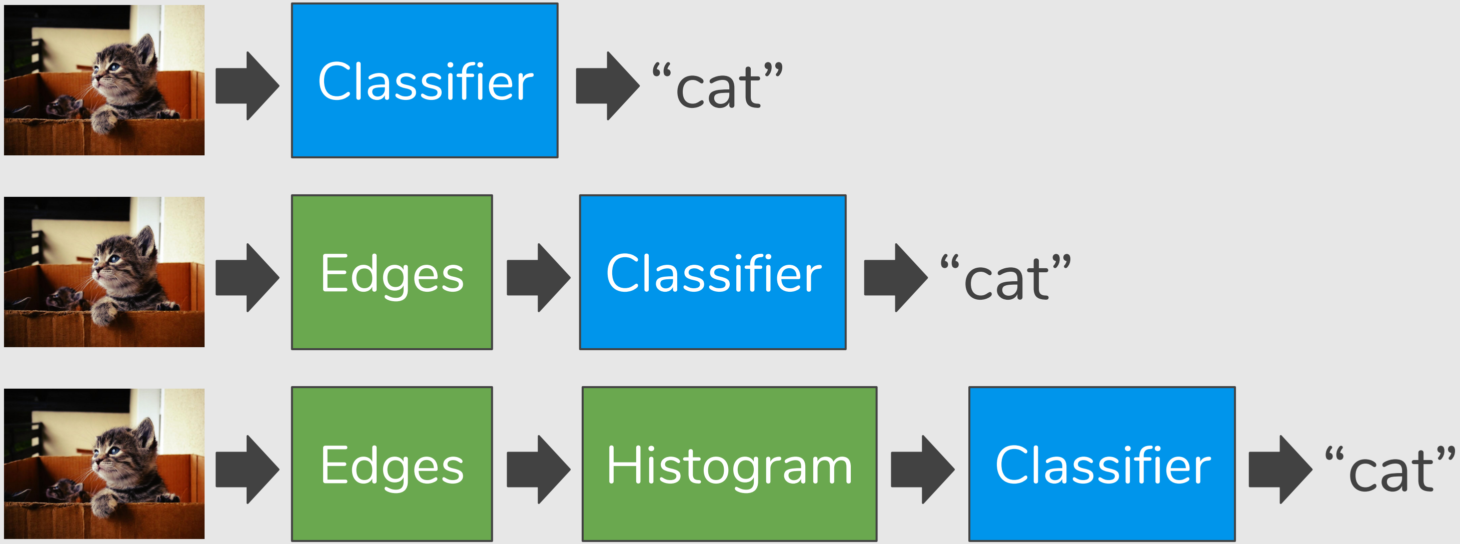


Traditional Recognition

Traditional Recognition



Traditional Recognition



Traditional Recognition

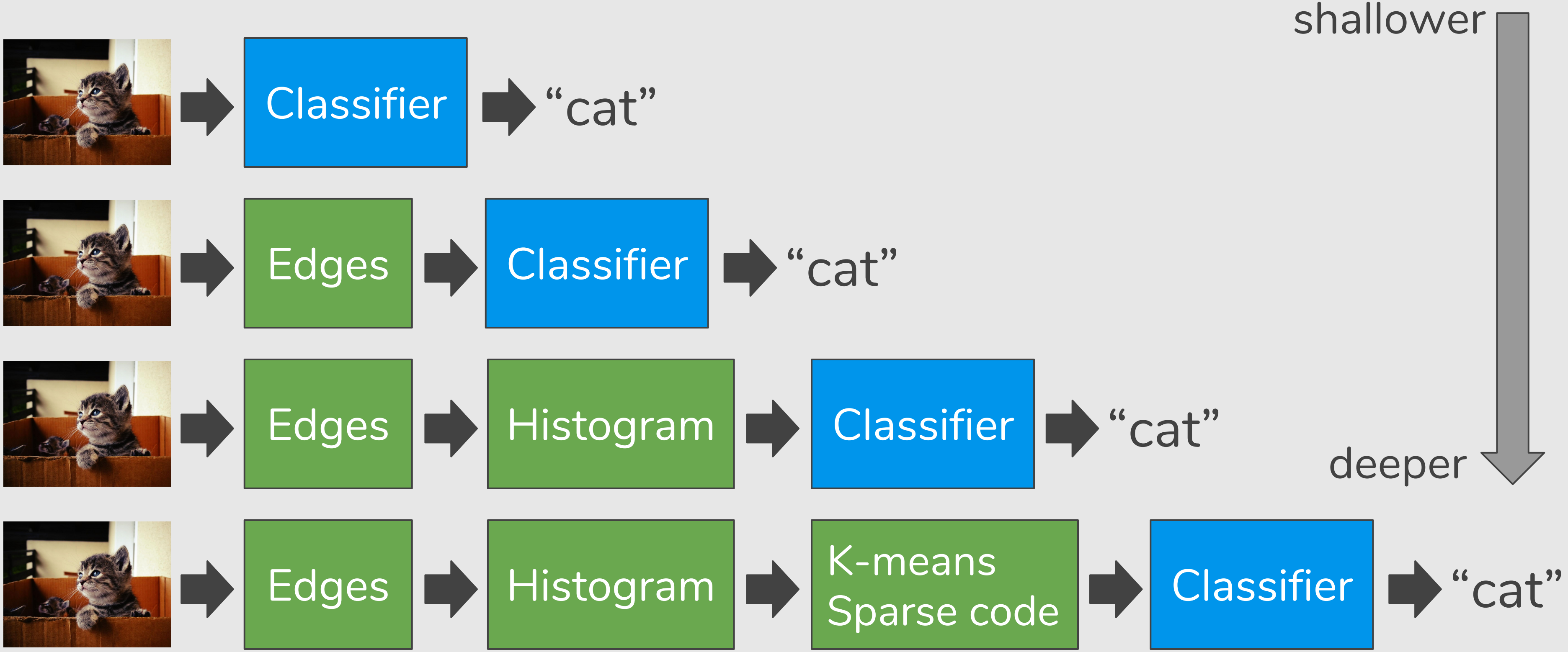
 → Classifier → “cat”

 → Edges → Classifier → “cat”

 → Edges → Histogram → Classifier → “cat”

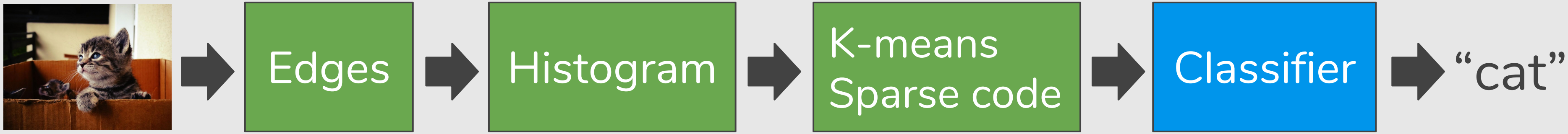
 → Edges → Histogram → K-means Sparse code → Classifier → “cat”

Traditional Recognition

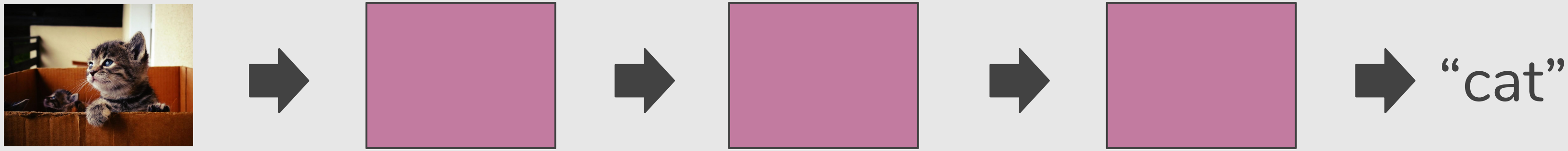


Deep Learning

Specialized components

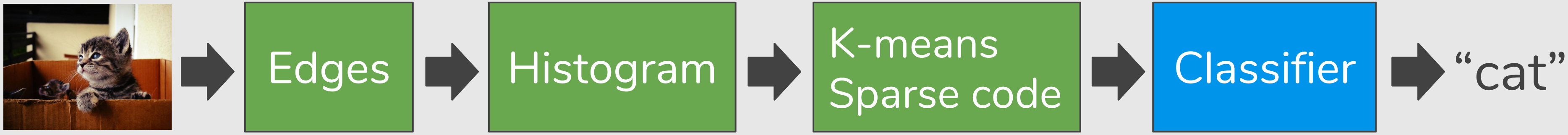


Generic components



Deep Learning

Specialized components



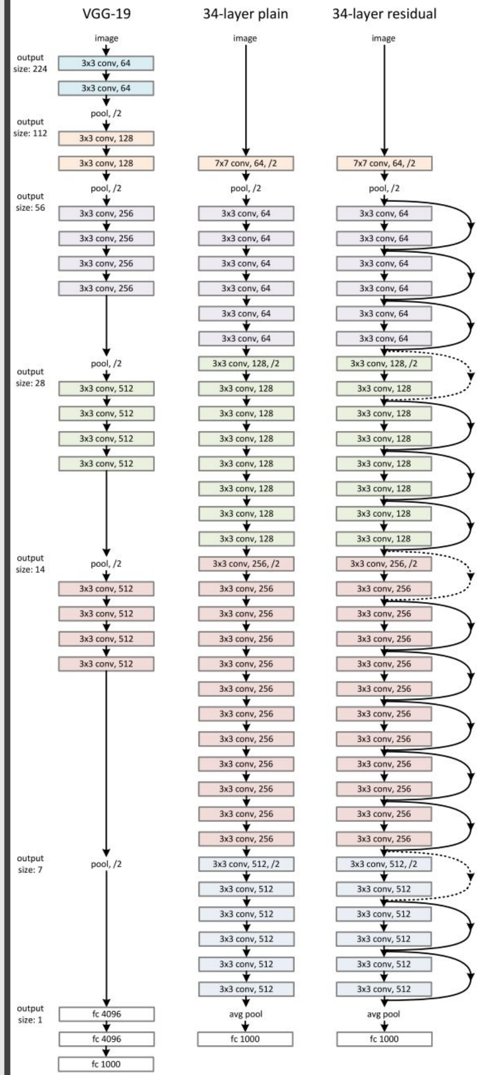
Generic components

Generic components, going deeper

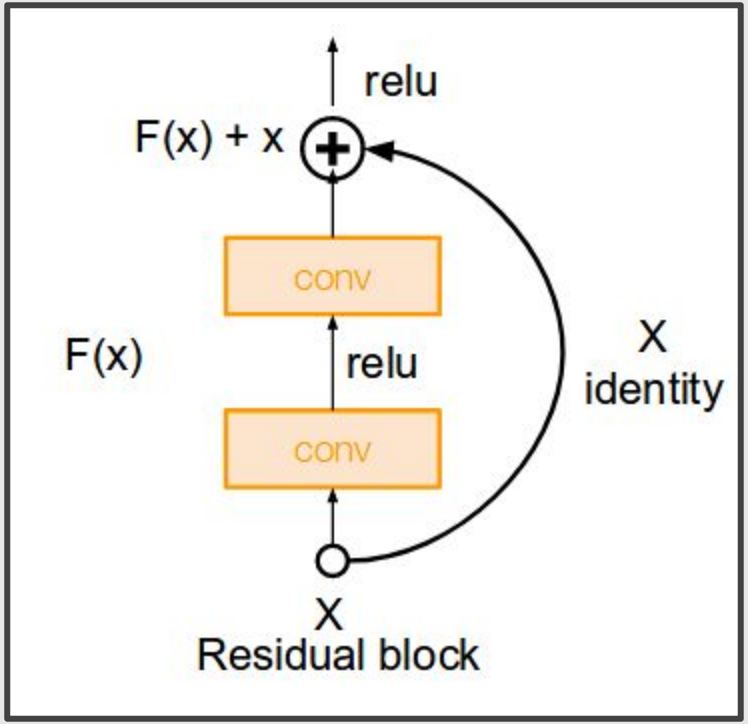
ResNet [He et al., 2015]

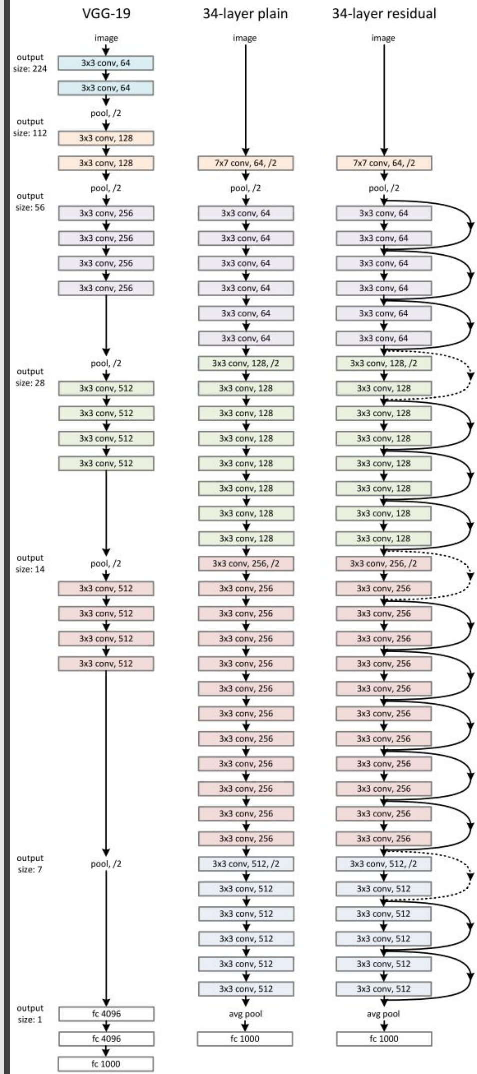
Very deep networks using residual connections

- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in



ResNet [He et al., 2015]



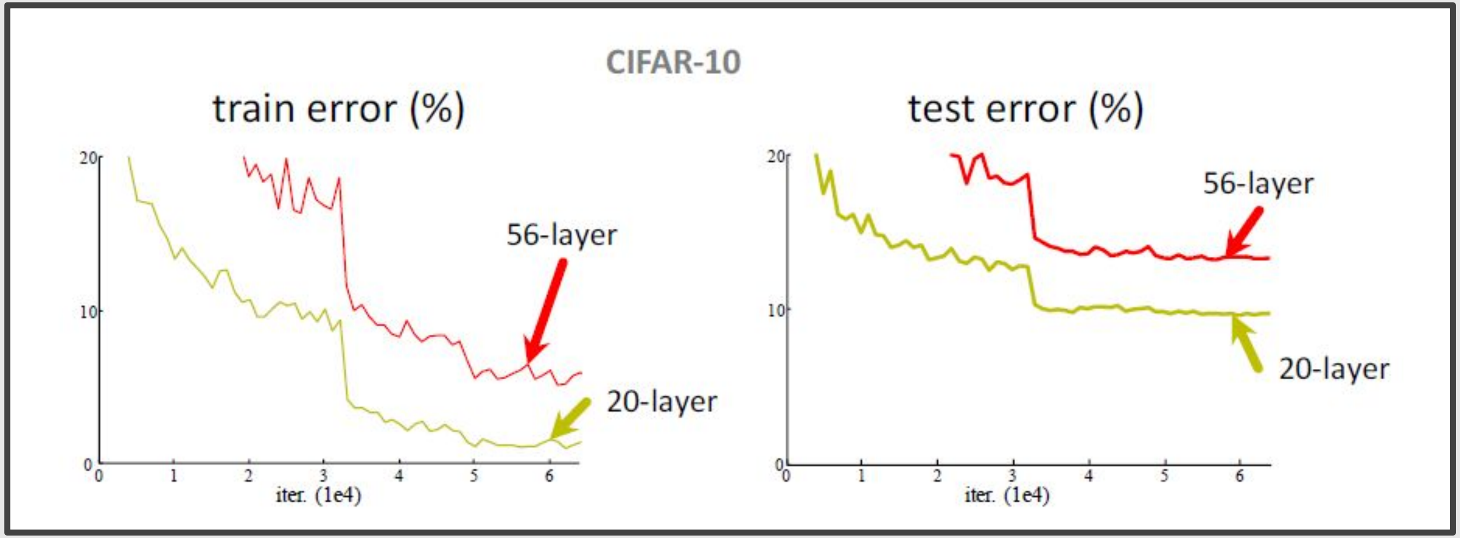


ResNet [He et al., 2015]

What happens when we continue stacking deeper layers
on a “plain” convolutional neural network?

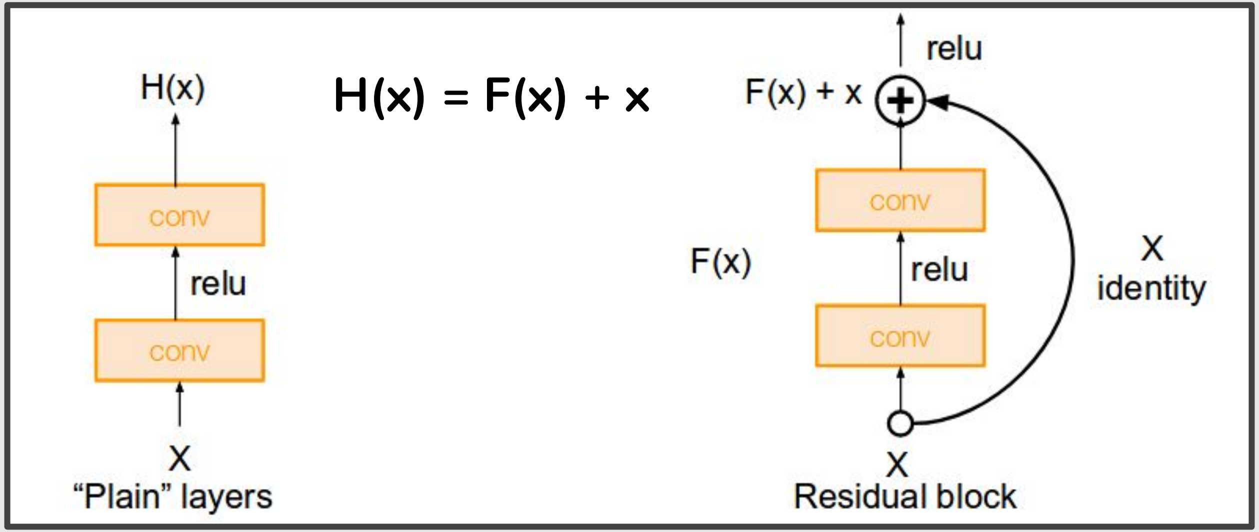
ResNet [He et al., 2015]

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?



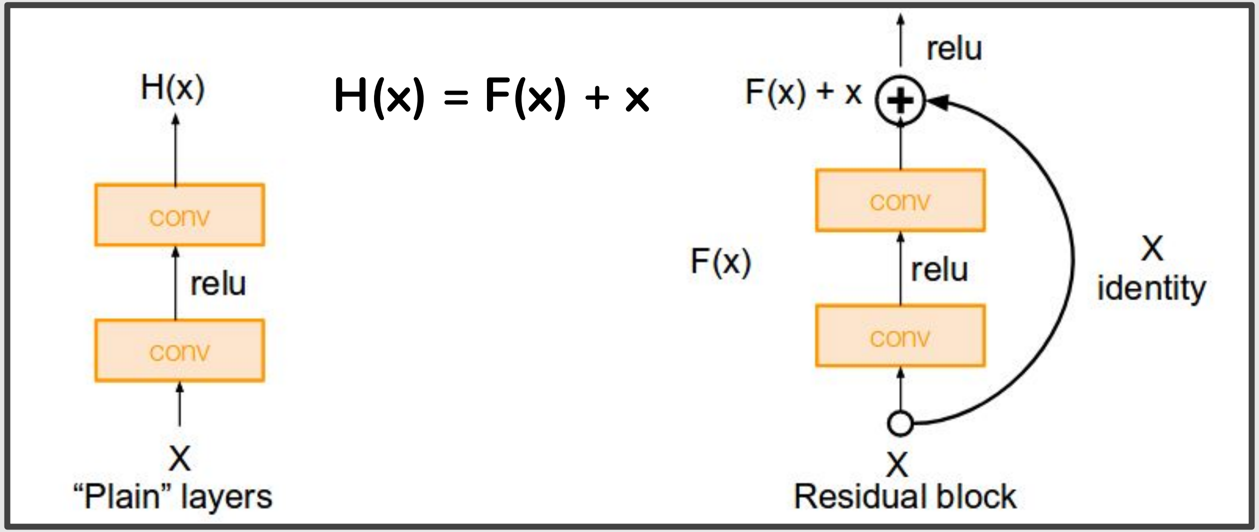
ResNet [He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



ResNet [He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping

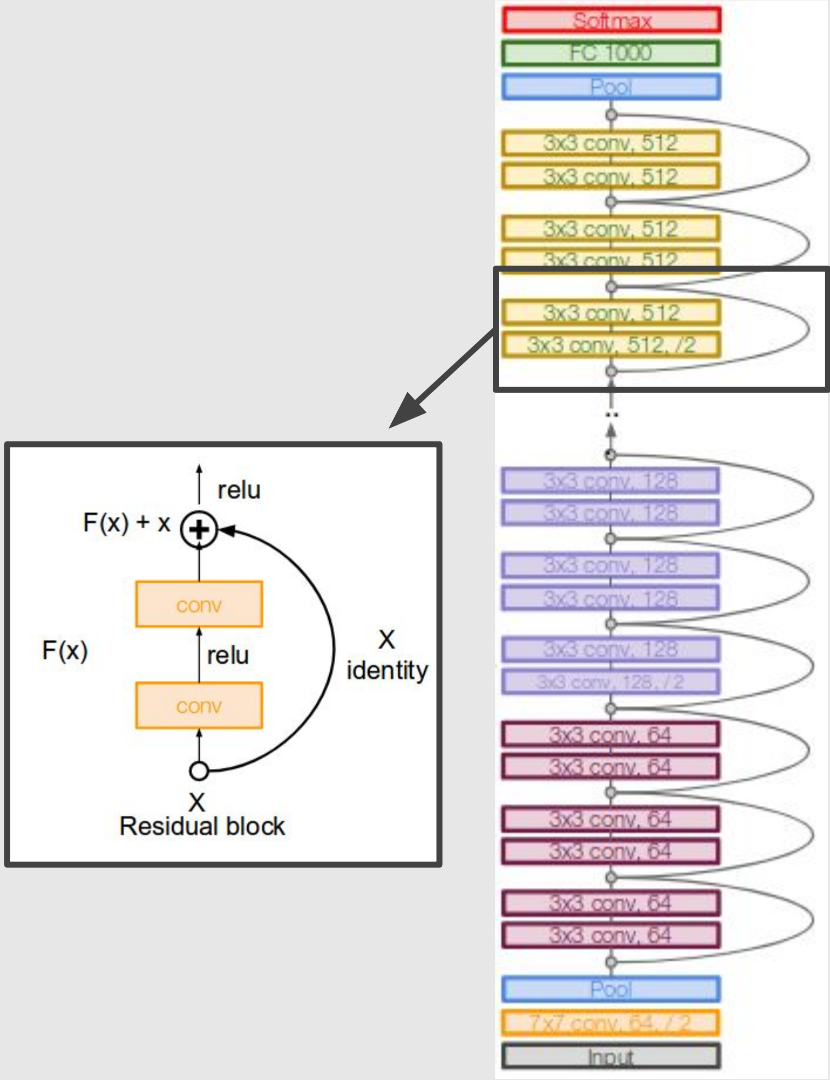


Use layers to fit residual F(x) = H(x) - x instead of H(x) directly

ResNet [He et al., 2015]

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)



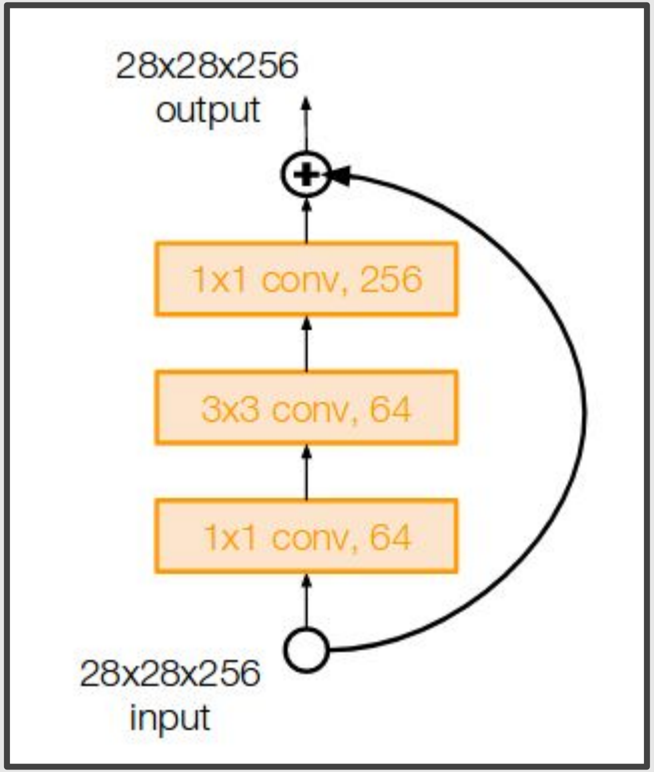
ResNet [He et al., 2015]

For deeper networks (**ResNet-50+**), use "bottleneck" layer to improve efficiency (similar to GoogLeNet)

1x1 conv, 256 filters projects back to 256 feature maps (28x28x256)

3x3 conv operates over only 64 feature maps

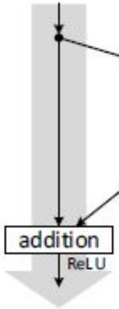
1x1 conv, 64 filters to project to 28x28x64



* ResNet-110 on CIFAR-10

$h(x) = x$

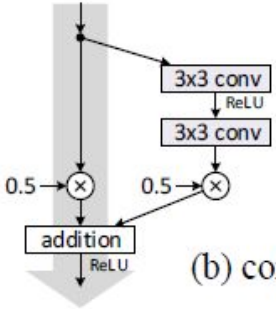
error: 6.6%



(a) original

$h(x) = 0.5x$

error: 12.4%

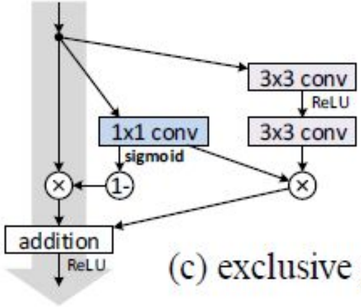


(b) constant scaling

$h(x) = \text{gate} \cdot x$

error: 8.7%

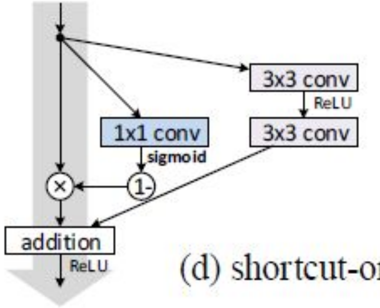
*similar to "Highway Network"



(c) exclusive gating

$h(x) = \text{gate} \cdot x$

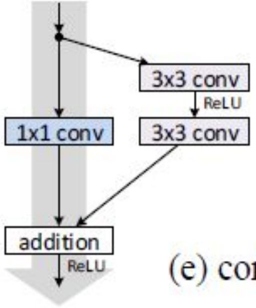
error: 12.9%



(d) shortcut-only gating

$h(x) = \text{conv}(x)$

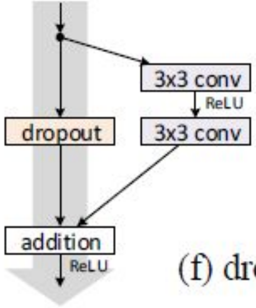
error: 12.2%



(e) conv shortcut

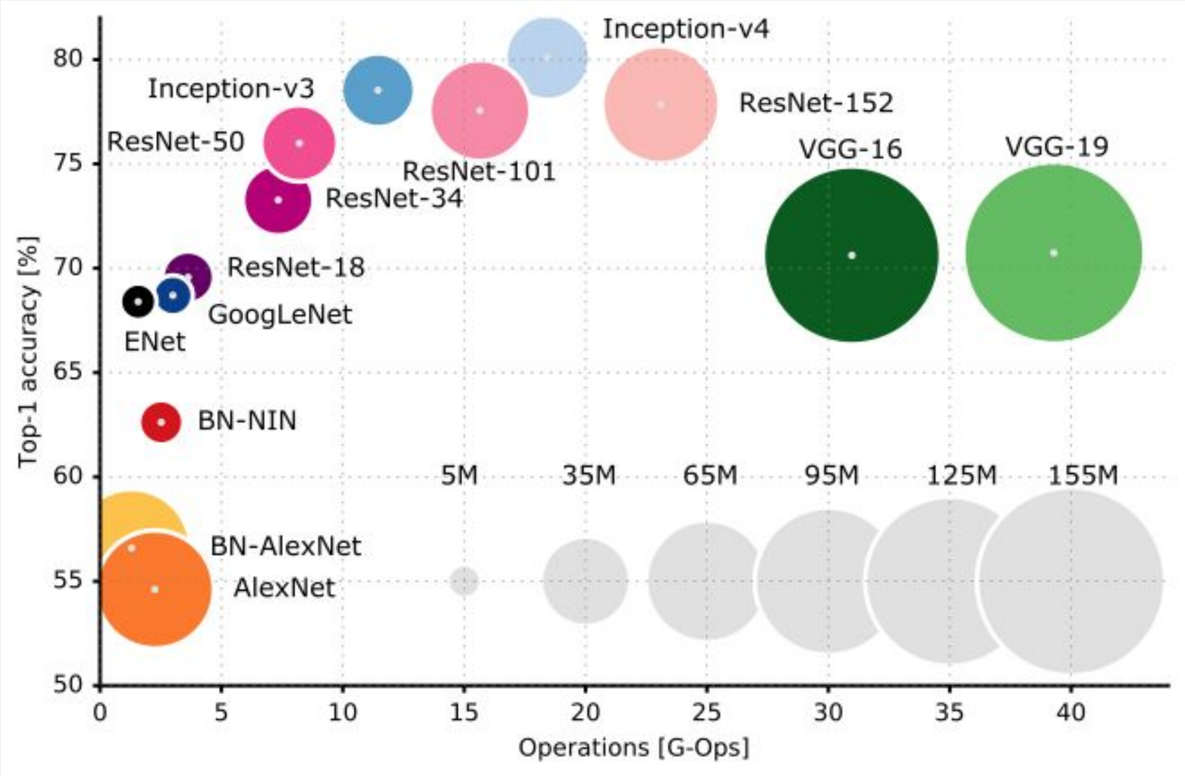
$h(x) = \text{dropout}(x)$

error: > 20%



(f) dropout shortcut

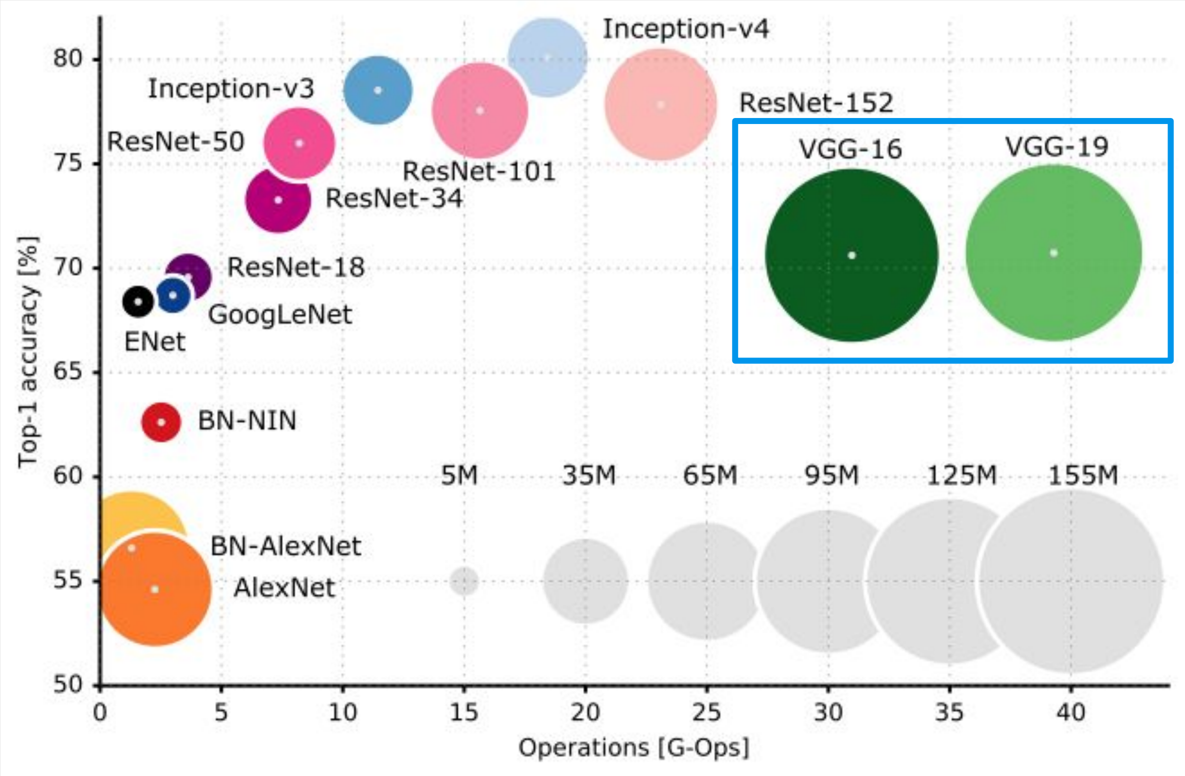
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

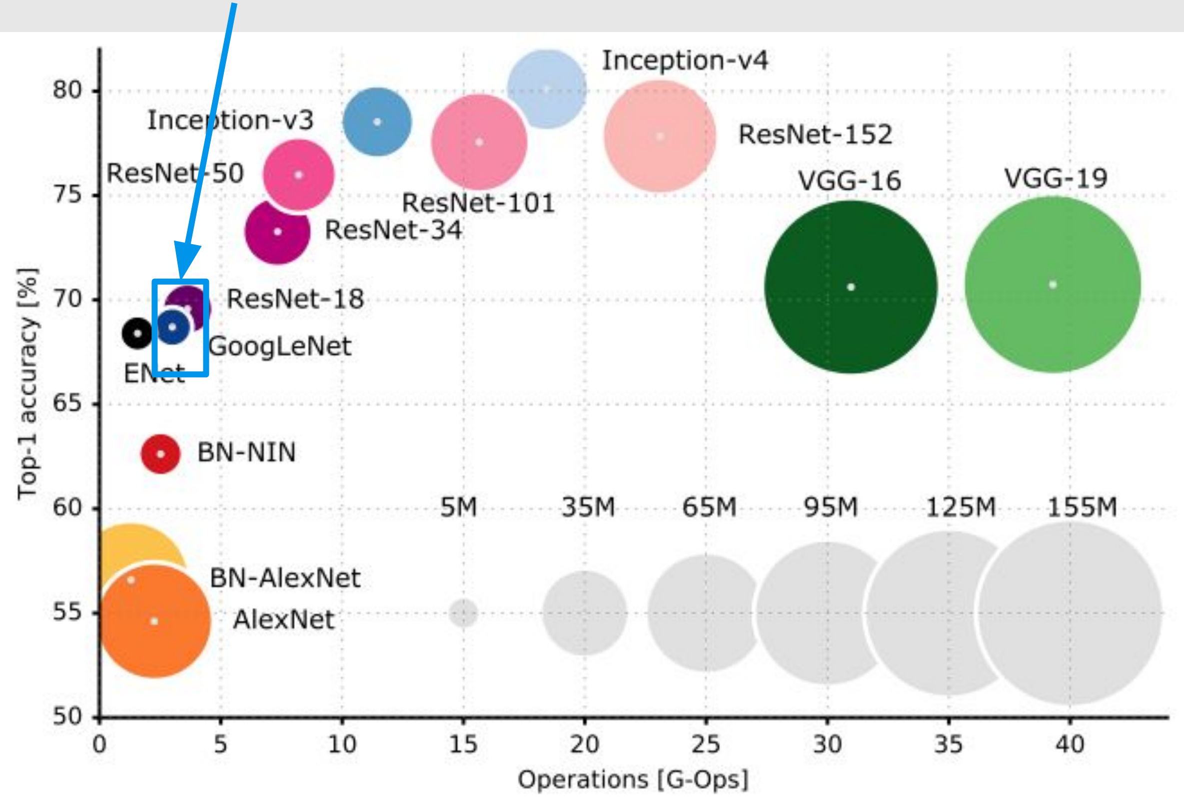
VGG: Highest memory, most operations



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

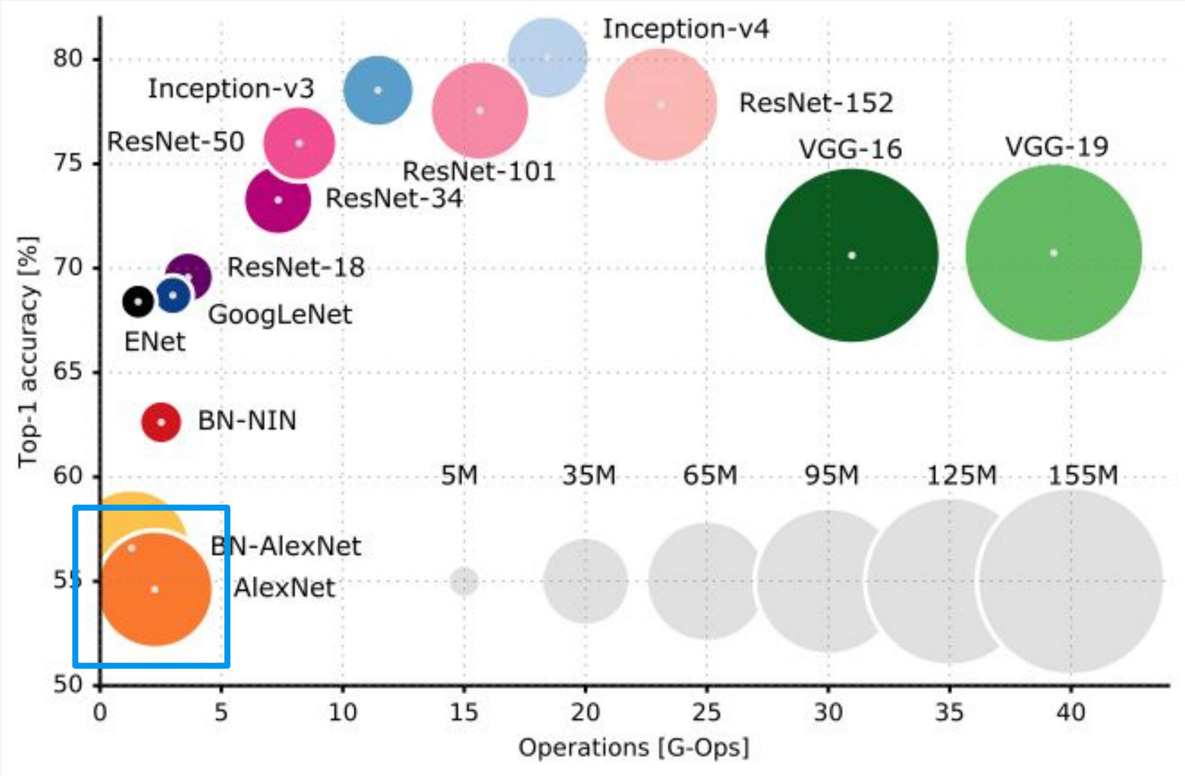
GoogLeNet: most efficient



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

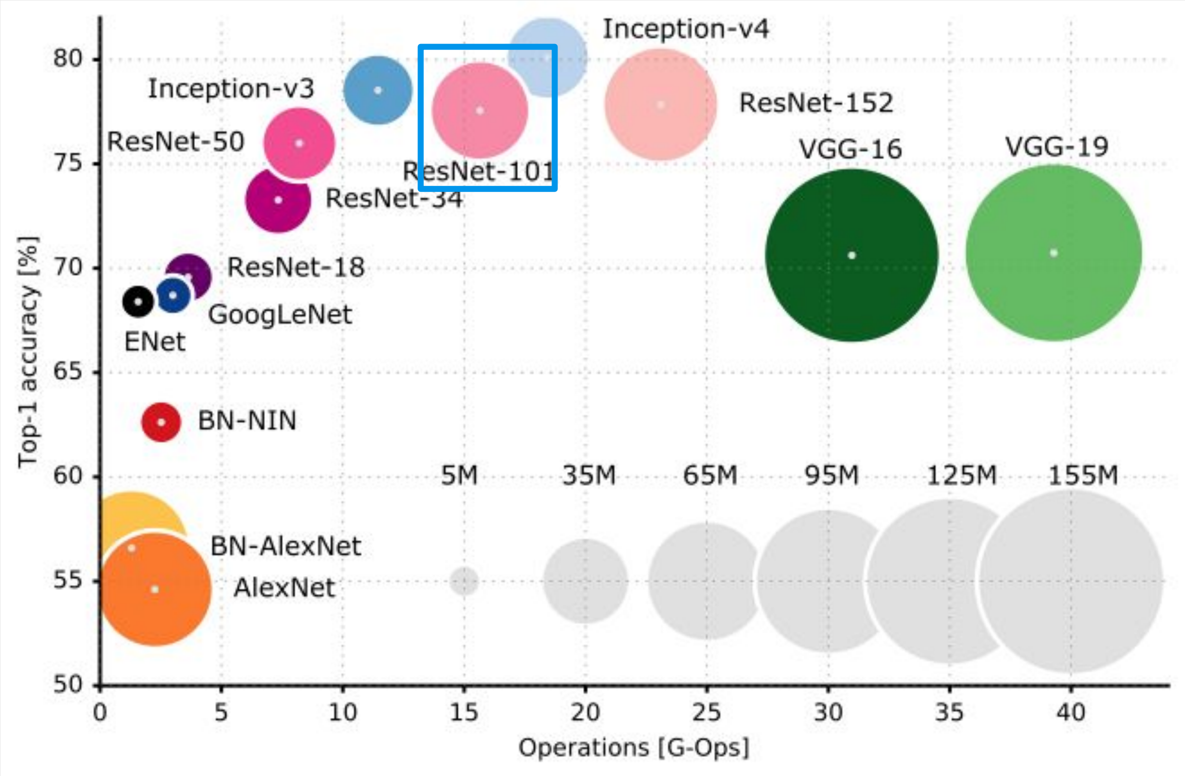
AlexNet: Smaller compute, still memory heavy, lower accuracy



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

ResNet: Moderate efficiency depending on model, highest accuracy



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

References

Machine Learning Books

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Chap. 11 & 13

Machine Learning Courses

- https://www.coursera.org/learn/neural-networks
- “The 3 popular courses on Deep Learning”: https://medium.com/towards-data-science/the-3-popular-courses-for-deeplearning-ai-ac37d4433bd