# CNN backbones

Vlad Shakhuro

16 October 2025

# Outline

1. CNN features and finetuning

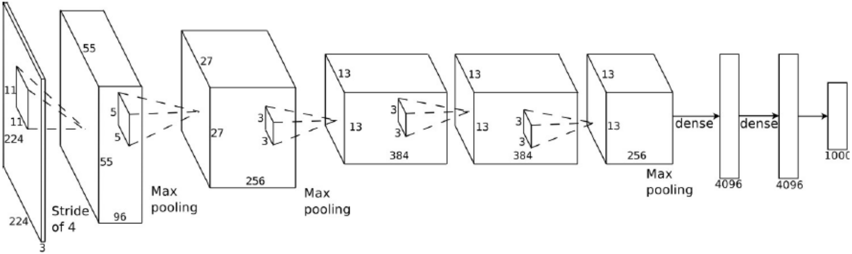
2. AlexNet, VGG, Inception

3. ResNet and its' improvements

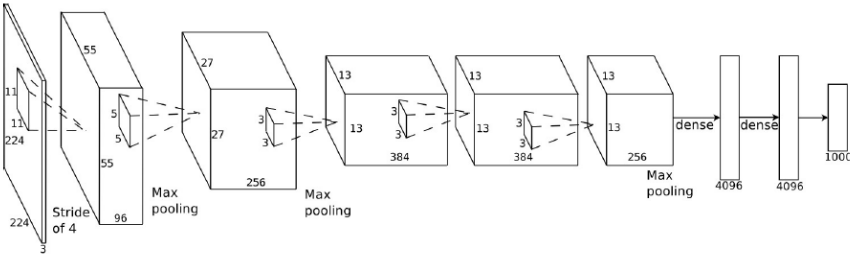
4. Mobile architectures

5. How good is ImageNet?

# How can we analyze a neural network?



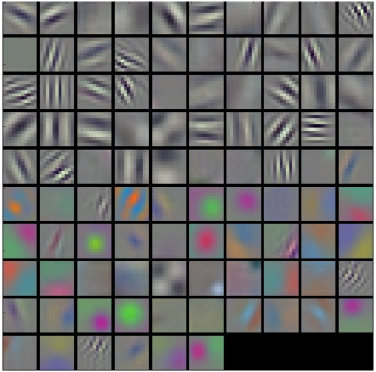
# How can we analyze a neural network?



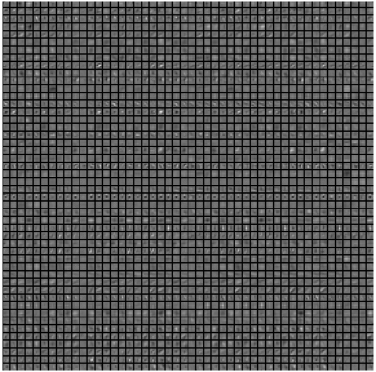

We can visualize:

- trained weights
- max activations of a particular neuron
- projection of a high-dimensional features space

# Visualizing filters



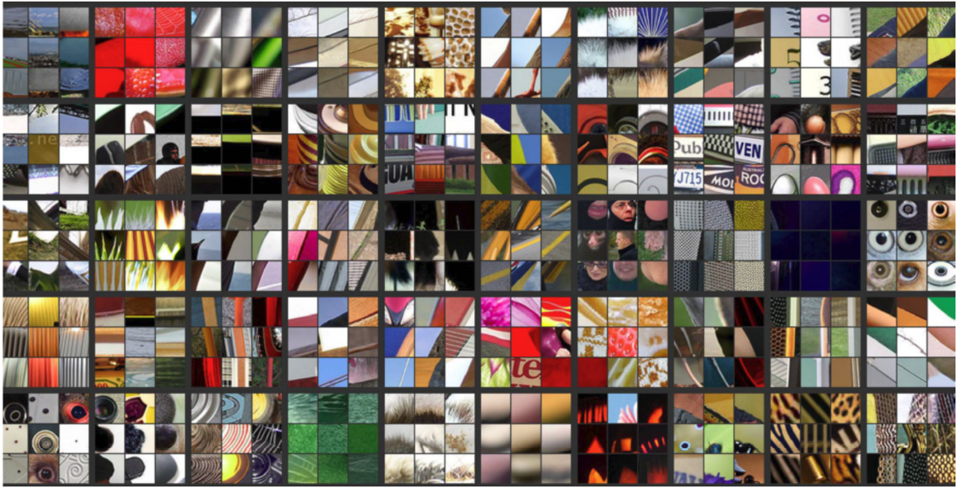
conv1



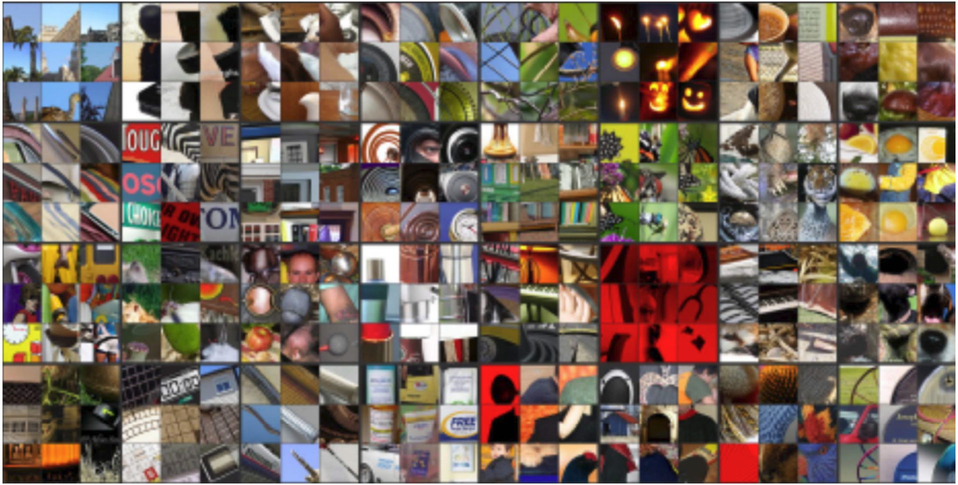
conv2

# Visualizing image fragments

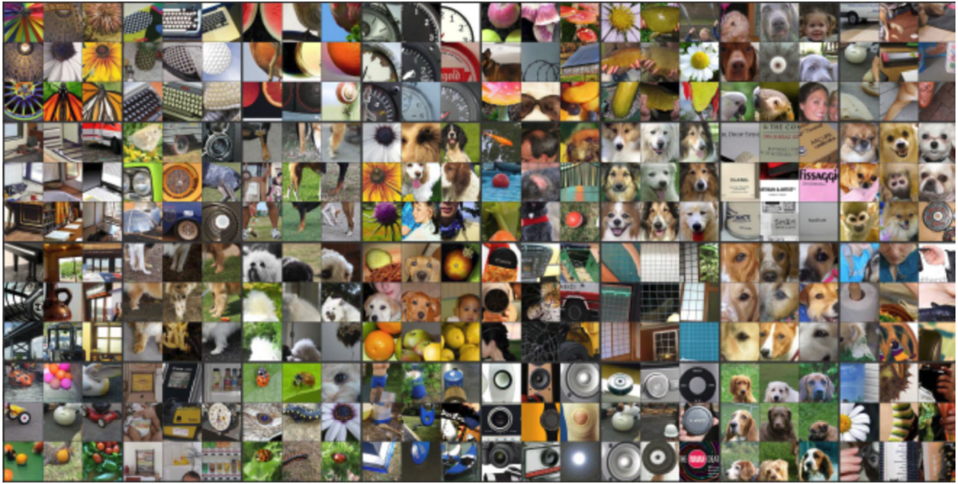
# Visualizing image fragments



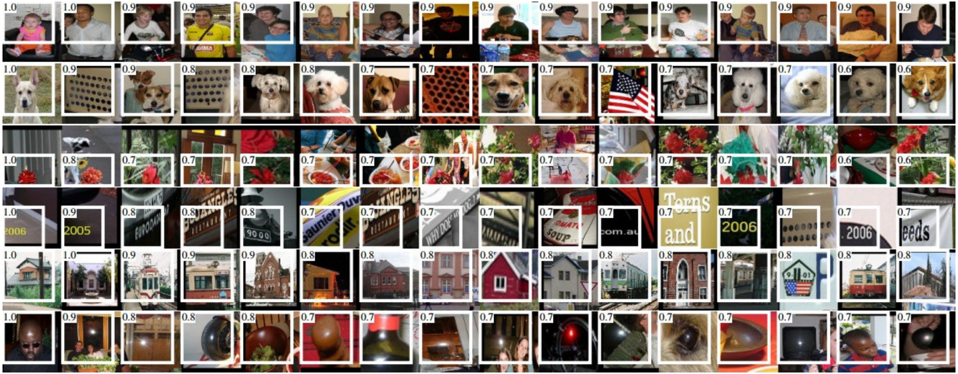
# Visualizing image fragments



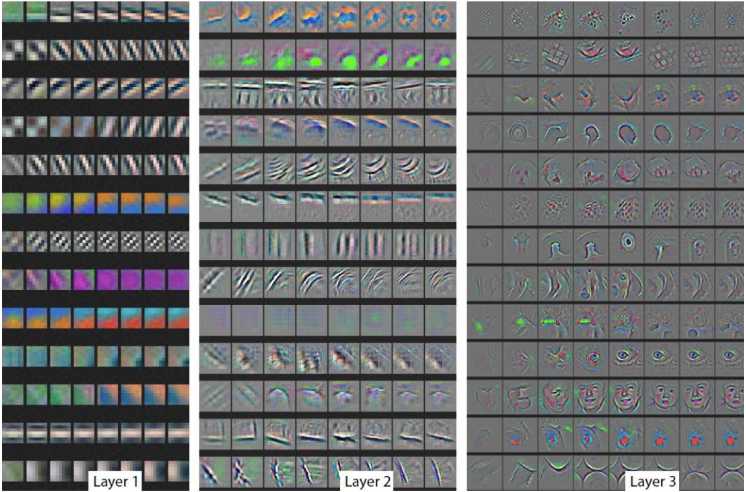
# Visualizing image fragments



# Visualizing image fragments



# Visualizing filters with deconvnet during training




Zeiler, Fergus. Visualizing and Understanding Convolutional Networks. ECCV 2014

# Visualizing feature space with t-SNE

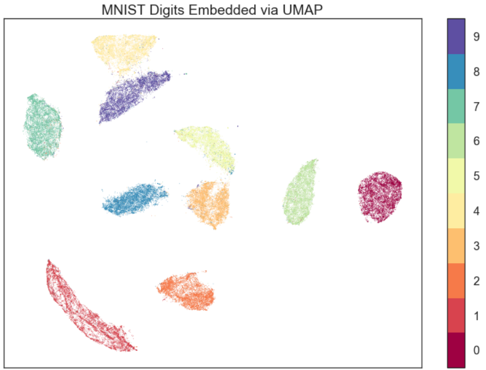
Compute $L_2$ distance for 4096-dim vectors (fc6 or fc7 layers)

Project in 2-dim space, approximately preserving $L_2$ distances

Visulize images. See that semantically similar images are close to each other

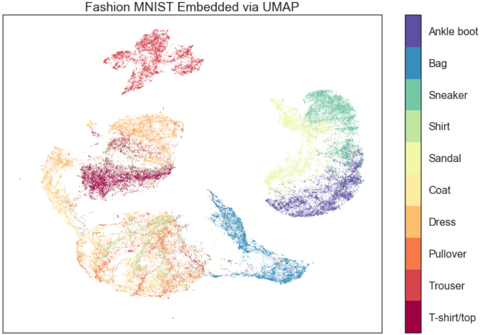
# Visualizing feature space with t-SNE

# Visualizing feature space with UMAP



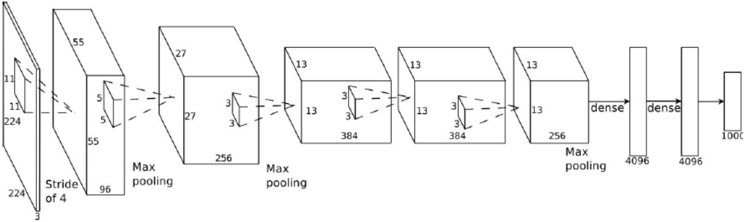

DMcInnes, Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv: 1802.03426

# Visualizing feature space with UMAP

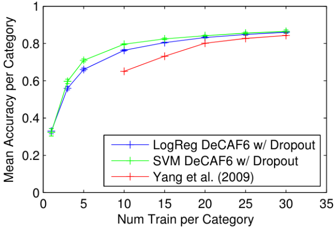



DMcInnes, Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv: 1802.03426

# Reusing features from classification networks

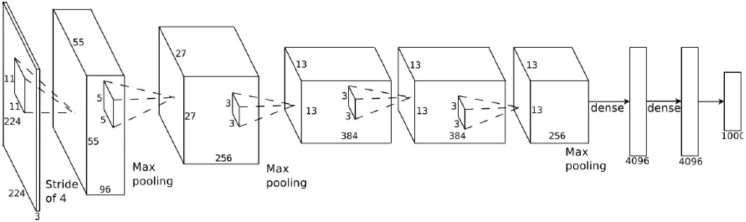



| | $DeCAF_5$ | $DeCAF_6$ | $DeCAF_7$ |
|---|---|---|---|
| LogReg | $63.29 \pm 6.6$ | $84.30 \pm 1.6$ | $84.87 \pm 0.6$ |
| LogReg with Dropout | - | $86.08 \pm 0.8$ | $85.68 \pm 0.6$ |
| SVM | $77.12 \pm 1.1$ | $84.77 \pm 1.2$ | $83.24 \pm 1.2$ |
| SVM with Dropout | - | $\mathbf{86.91 \pm 0.7}$ | $85.51 \pm 0.9$ |
| Yang et al. (2009) | | 84.3 | |
| Jarrett et al. (2009) | | 65.5 | |



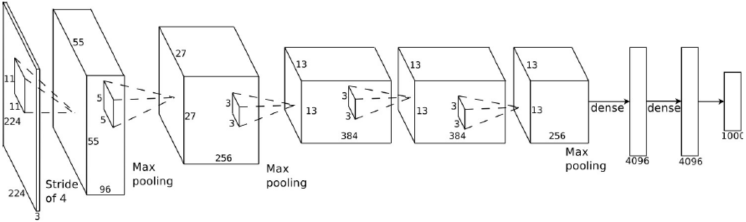

Donahue et al. Decaf: A deep convolutional activation feature for generic visual recognition. ICLR 2014

# Finetuning a neural network




Replace last classifier layer and finetune the network with smaller learning rate. During finetuning we may use small training dataset

# Finetuning a neural network




Replace last classifier layer and finetune the network with smaller learning rate. During finetuning we may use small training dataset

We now come to idea of **backbones**: baseline architectures that are pretrained on large datasets

# Outline

1. CNN features and finetuning

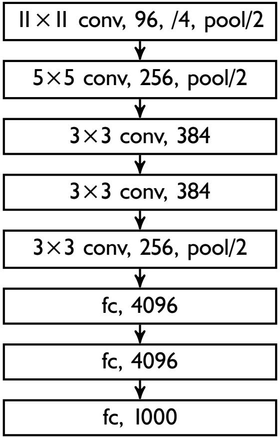
2. AlexNet, VGG, Inception

3. ResNet and its' improvements

4. Mobile architectures

5. How good is ImageNet?

# AlexNet




Krizhevsky et al. Imagenet classification with deep convolutional neural networks. NIPS 2012

# Applying AlexNet to different resolutions

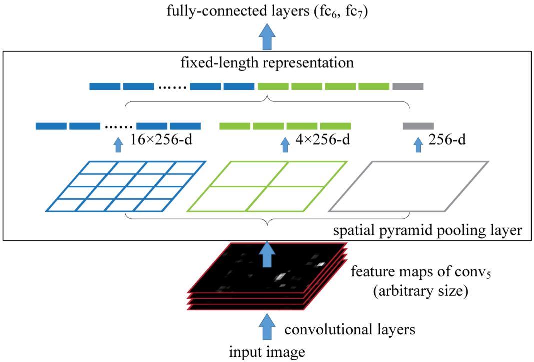
- Fixed resolution:

crop

warp

- Sample several random crops, average results
- Scan whole image with fixed size window, average scores

# Spatial Pyramid Pooling




Single pooling layer across all features is called **average pooling**

He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. TPAMI 2015

# VGG

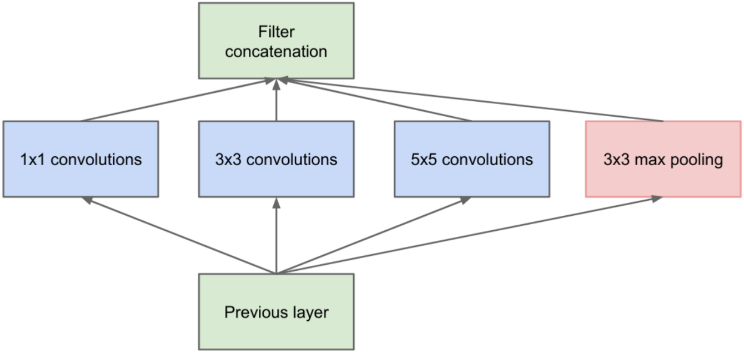
Key ideas:

- Use only 3×3 convolutions
- Increase depth
- Use only pooling for decreasing resolution
- Increase #filters in 2 times after pooling

| A | A-LRN | B | C | D | E |
|---|---|---|---|---|---|
| 11 weight layers | 11 weight layers | 13 weight layers | 16 weight layers | 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | | | | | |
| conv3-64 | conv3-64<br>**LRN** | conv3-64<br>**conv3-64** | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 |
| maxpool | | | | | |
| conv3-128 | conv3-128 | conv3-128<br>**conv3-128** | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 |
| maxpool | | | | | |
| conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256<br>**conv1-256** | conv3-256<br>conv3-256<br>**conv3-256** | conv3-256<br>conv3-256<br>conv3-256<br>**conv3-256** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| FC-4096 | | | | | |
| FC-4096 | | | | | |
| FC-1000 | | | | | |
| soft-max | | | | | |

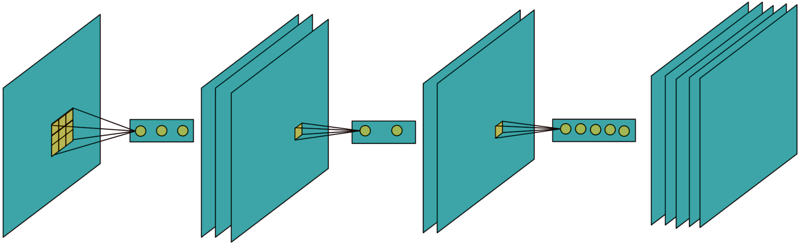
Simonyan, Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015

# Inception block




Szegedy et al. Going deeper with convolutions. CVPR 2015

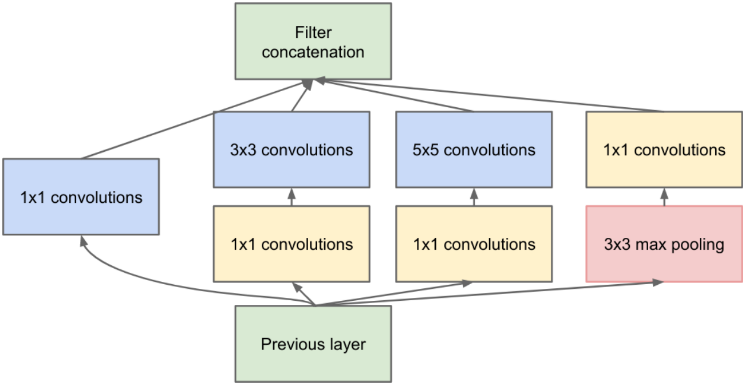
# 1 × 1 convolutions



1 × 1 convolution maps $N_{\text{in}}$ channels to $N_{\text{out}}$ channels.
May be used as:

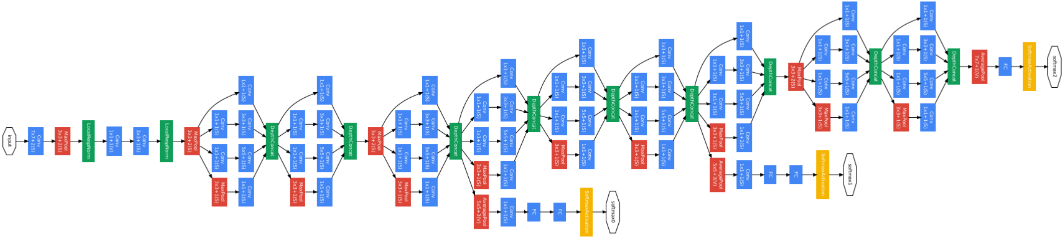
- a set of local classifiers
- a method for expanding ($N_{\text{in}} < N_{\text{out}}$) or reducing ($N_{\text{in}} > N_{\text{out}}$) tensor depth

# Inception block with dim reduction




Szegedy et al. Going deeper with convolutions. CVPR 2015

# Inception architecture



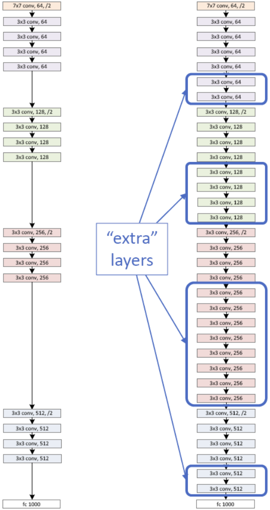
Deep network made of inception blocks. To make training more stable, uses several heads for supervision

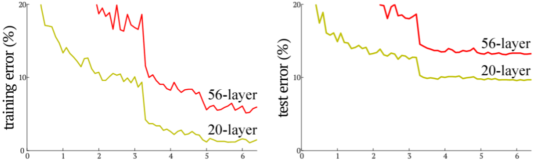
Szegedy et al. Going deeper with convolutions. CVPR 2015

# Outline

1. CNN features and finetuning
2. AlexNet, VGG, Inception
3. ResNet and its' improvements
4. Mobile architectures
5. How good is ImageNet?

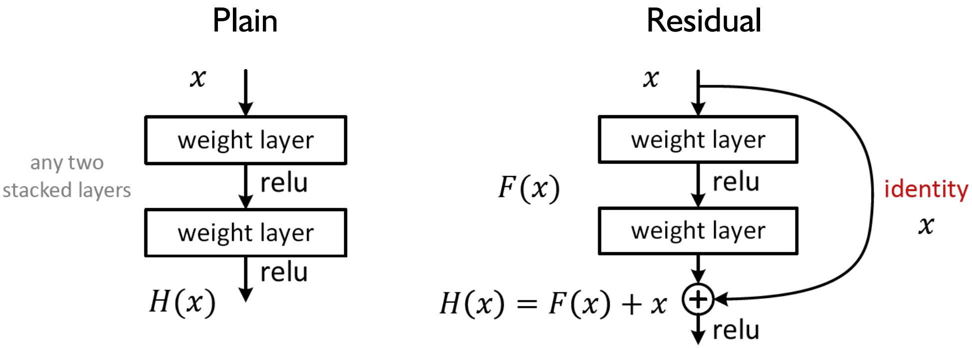
# Increasing network depth



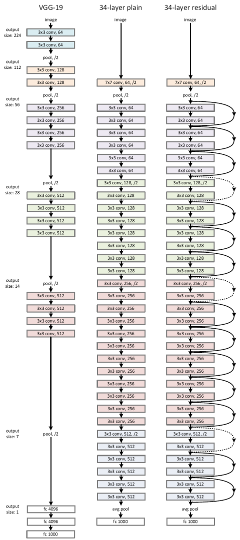

Simply increasing network depth doesn’t work. However using identity layers we may obtain neural network of arbitrary depth. Therefore it’s training problem

# Residual block



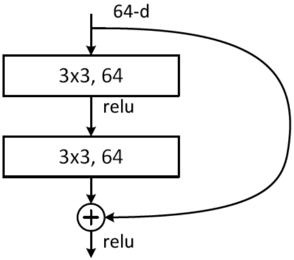

Skip connections will help network learn additive component to the identity function. Gradient are able now to flow through skip connections

# ResNet

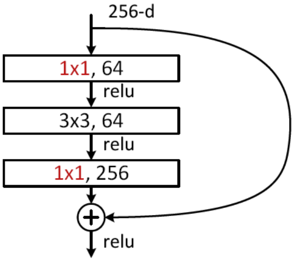



Only 3×3 convolutions, subsampling using stride 2

Repeating residual bottleneck blocks:



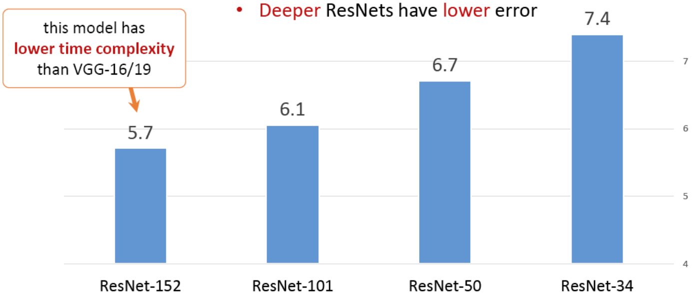

all-3x3

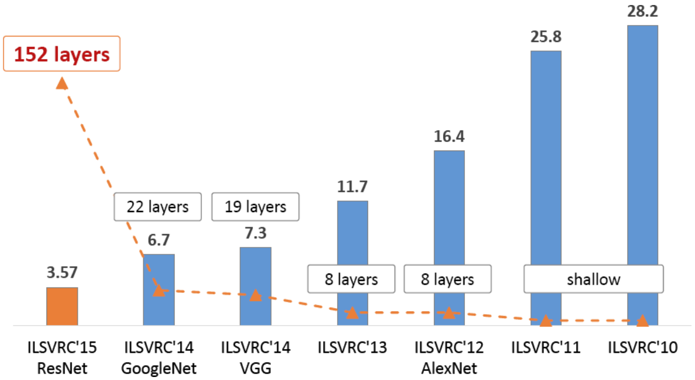

bottleneck

(for ResNet-50/101/152)

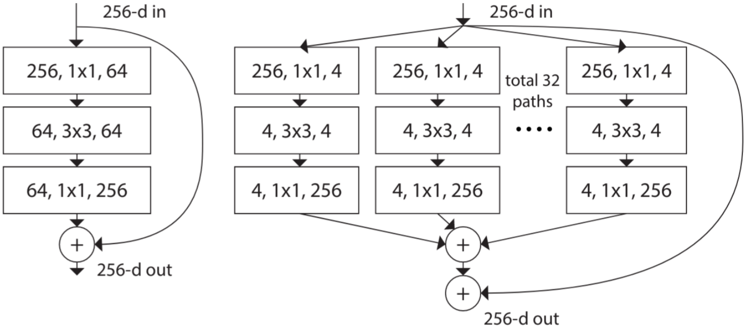
# ResNet results



# Comparing ResNet to previous backbones

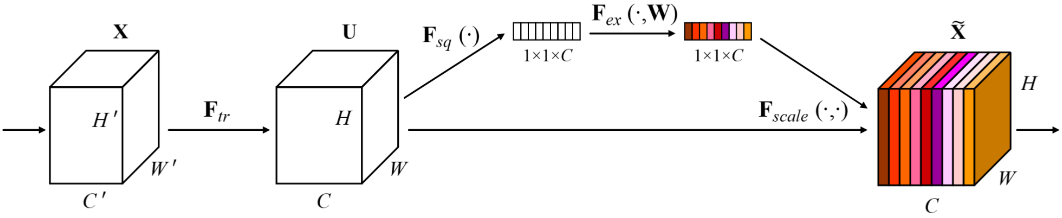


# ResNeXt




Xie et al. Aggregated Residual Transformations for Deep Neural Networks. CVPR 2017

# Squeeze-and-Excitation




| | original | | re-implementation | | | SENet | | |
|---|---|---|---|---|---|---|---|---|
| | top-1 err. | top-5 err. | top-1err. | top-5 err. | GFLOPs | top-1 err. | top-5 err. | GFLOPs |
| ResNet-50 [10] | 24.7 | 7.8 | 24.80 | 7.48 | 3.86 | $23.29_{(1.51)}$ | $6.62_{(0.86)}$ | 3.87 |
| ResNet-101 [10] | 23.6 | 7.1 | 23.17 | 6.52 | 7.58 | $22.38_{(0.79)}$ | $6.07_{(0.45)}$ | 7.60 |
| ResNet-152 [10] | 23.0 | 6.7 | 22.42 | 6.34 | 11.30 | $21.57_{(0.85)}$ | $5.73_{(0.61)}$ | 11.32 |
| ResNeXt-50 [47] | 22.2 | - | 22.11 | 5.90 | 4.24 | $21.10_{(1.01)}$ | $5.49_{(0.41)}$ | 4.25 |
| ResNeXt-101 [47] | 21.2 | 5.6 | 21.18 | 5.57 | 7.99 | $20.70_{(0.48)}$ | $5.01_{(0.56)}$ | 8.00 |
| VGG-16 [39] | - | - | 27.02 | 8.81 | 15.47 | $25.22_{(1.80)}$ | $7.70_{(1.11)}$ | 15.48 |
| BN-Inception [16] | 25.2 | 7.82 | 25.38 | 7.89 | 2.03 | $24.23_{(1.15)}$ | $7.14_{(0.75)}$ | 2.04 |
| Inception-ResNet-v2 [42] | $19.9^{\dagger}$ | $4.9^{\dagger}$ | 20.37 | 5.21 | 11.75 | $19.80_{(0.57)}$ | $4.79_{(0.42)}$ | 11.76 |

Hu et al. Squeeze-and-Excitation Networks. CVPR 2018

# Outline

1. CNN features and finetuning

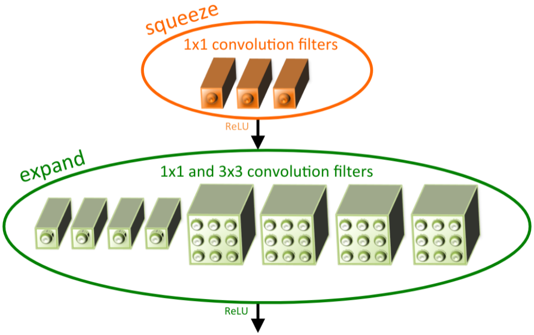
2. AlexNet, VGG, Inception

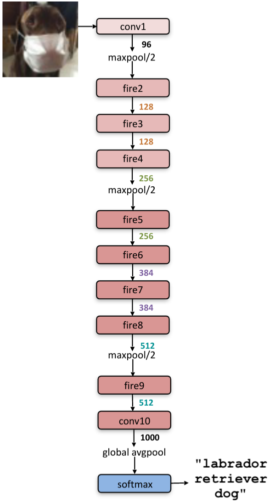
3. ResNet and its' improvements

4. Mobile architectures

5. How good is ImageNet?

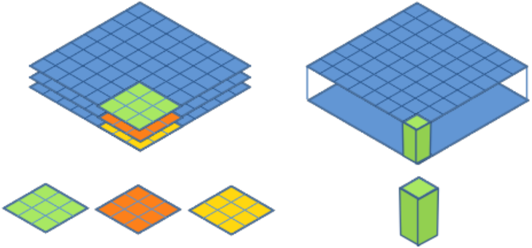
# SqueezeNet




Iandola et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. ICLR 2017

# Depthwise separable convolutions



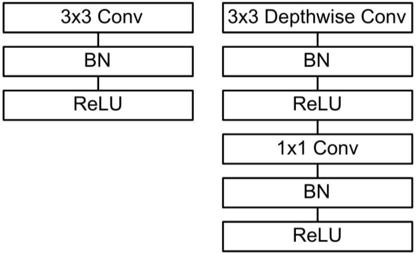
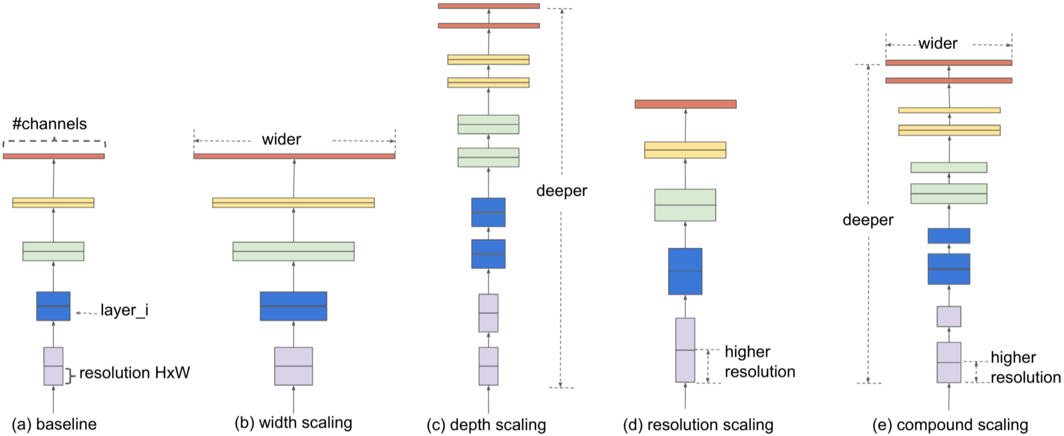



Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

Howard et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017

# EfficientNet




Tan, Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICLR 2019

# EfficientNet

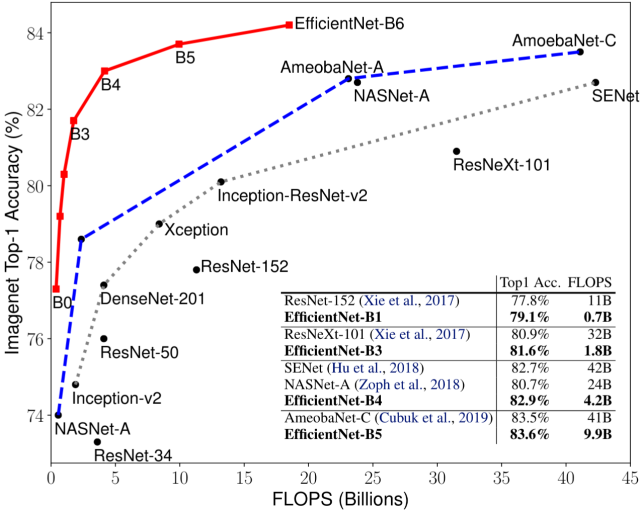
$$\text{depth: } d = \alpha^\phi$$
$$\text{width: } w = \beta^\phi$$
$$\text{resolution: } r = \gamma^\phi$$
$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

| Stage $i$ | Operator $\hat{\mathcal{F}}_i$ | Resolution $\hat{H}_i \times \hat{W}_i$ | #Channels $\hat{C}_i$ | #Layers $\hat{L}_i$ |
|---|---|---|---|---|
| 1 | Conv3x3 | $224 \times 224$ | 32 | 1 |
| 2 | MBConv1, k3x3 | $112 \times 112$ | 16 | 1 |
| 3 | MBConv6, k3x3 | $112 \times 112$ | 24 | 2 |
| 4 | MBConv6, k5x5 | $56 \times 56$ | 40 | 2 |
| 5 | MBConv6, k3x3 | $28 \times 28$ | 80 | 3 |
| 6 | MBConv6, k5x5 | $14 \times 14$ | 112 | 3 |
| 7 | MBConv6, k5x5 | $14 \times 14$ | 192 | 4 |
| 8 | MBConv6, k3x3 | $7 \times 7$ | 320 | 1 |
| 9 | Conv1x1 & Pooling & FC | $7 \times 7$ | 1280 | 1 |

EfficientNet-B0

Tan, Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICLR 2019

# EfficientNet results




| | Top1 Acc. | FLOPS |
|---|---|---|
| ResNet-152 (Xie et al., 2017) | 77.8% | 11B |
| **EfficientNet-B1** | **79.1%** | **0.7B** |
| ResNeXt-101 (Xie et al., 2017) | 80.9% | 32B |
| **EfficientNet-B3** | **81.6%** | **1.8B** |
| SENet (Hu et al., 2018) | 82.7% | 42B |
| NASNet-A (Zoph et al., 2018) | 80.7% | 24B |
| **EfficientNet-B4** | **82.9%** | **4.2B** |
| AmeobaNet-C (Cubuk et al., 2019) | 83.5% | 41B |
| **EfficientNet-B5** | **83.6%** | **9.9B** |

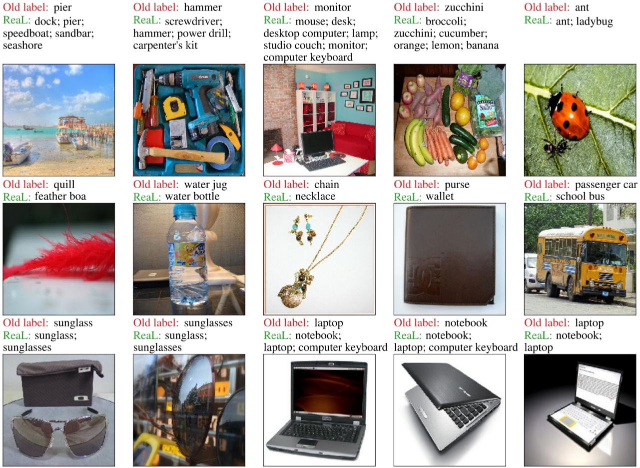
Tan, Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICLR 2019

# Outline

1. CNN features and finetuning
2. AlexNet, VGG, Inception
3. ResNet and its' improvements
4. Mobile architectures
5. How good is ImageNet?

# Classification example

# Relabelling ImageNet




Beyer et al. Are we done with ImageNet? 2020

# Relabelling ImageNet

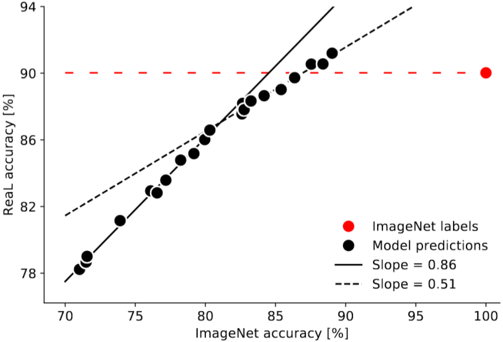



Figure 4: Comparing progress on ReaL accuracy and the original ImageNet accuracy. We measured the association between both metrics by regressing ImageNet accuracy onto ReaL accuracy for the first (solid line) and second half (dashed line) of the models in our pool.

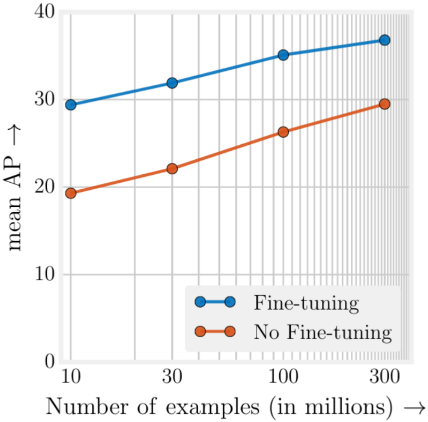
Beyer et al. Are we done with ImageNet? 2020

# Relabelling ImageNet

| | Model | ImageNet accuracy | | | ReaL accuracy | | |
|---|---|---|---|---|---|---|---|
| | | 90 epochs | 270 epochs | 900 epochs | 90 epochs | 270 epochs | 900 epochs |
| ResNet-50 | Baseline | 76.0 | 76.9 (+0.9) | 75.9 (-0.1) | 82.5 | 82.9 (+0.4) | 81.6 (-0.9) |
| | + Sigmoid | 76.3 (+0.3) | 77.8 (+1.8) | 76.9 (+0.9) | 83.0 (+0.5) | 83.9 (+1.4) | 82.7 (+0.2) |
| | + Clean | 76.4 (+0.4) | 77.8 (+1.8) | 77.4 (+1.4) | 82.8 (+0.3) | 83.7 (+1.2) | 83.3 (+0.8) |
| | + Both | 76.6 (+0.6) | 78.2 (+2.2) | **78.5** (+2.5) | 83.1 (+0.6) | **84.3** (+1.8) | 84.1 (+1.6) |
| ResNet-152 | Baseline | 78.0 | 78.3 (+0.3) | 77.1 (-0.9) | 84.1 | 83.8 (-0.3) | 82.3 (-1.8) |
| | + Sigmoid | 78.5 (+0.5) | 78.7 (+0.7) | 77.4 (-0.6) | 84.6 (+0.5) | 84.3 (+0.2) | 82.7 (-1.4) |
| | + Clean | 78.6 (+0.6) | 79.6 (+1.6) | 79.0 (+1.0) | 84.4 (+0.3) | 85.0 (+0.9) | 84.4 (+0.3) |
| | + Both | 78.7 (+0.7) | **79.8** (+1.8) | 79.3 (+1.3) | 84.6 (+0.5) | **85.2** (+1.1) | 84.5 (+0.4) |

Beyer et al. Are we done with ImageNet? 2020

# Using larger datasets




- JFT-300M dataset (Google)
- 1B tags, 18291 classes
- 375M tags after filtering, ~20% errors
- Training ResNet-101 on 50×K80 for a month

Sun et al. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. ICCV 2017

# Conclusion

We reviewed foolowing topics:

- using backbones as universal feature extractors
- building deep networks using basic blocks from $3 \times 3$ convolutions
- using multiple paths for processing tensors
- skip connections for training very deep networks
- basic attention mechanism
- factorizing convolutions
- ImageNet quality and larger datasets