Computer Vision with Deep Learning: an overview

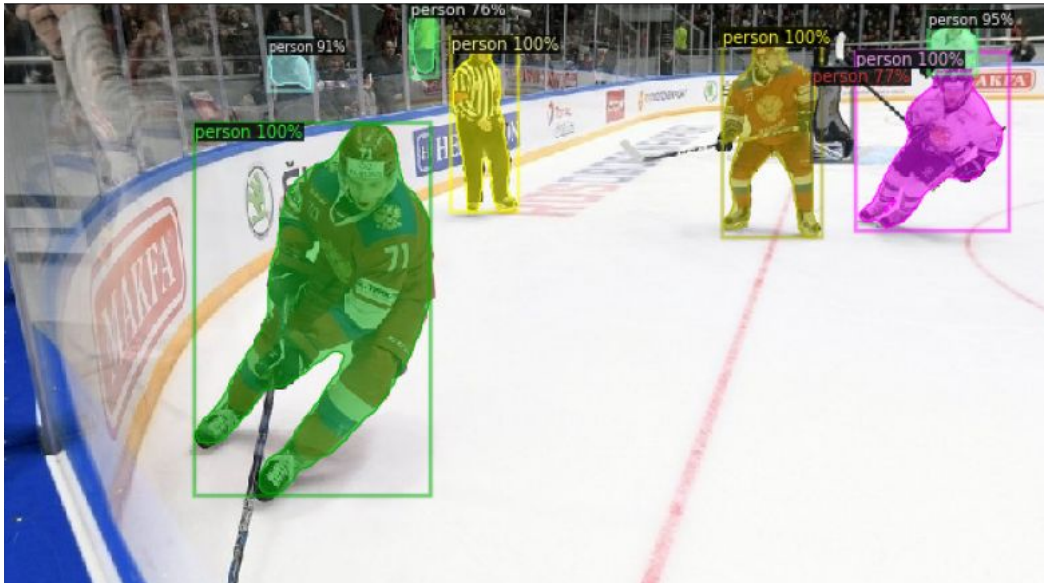
Radoslav Neychev

Outline

- Different problems in Computer Vision
- Convolutional layer recap
- Main ideas of famous architectures in CV
- Datasets overview

Different problems in Computer Vision – an overview

Computer Vision tasks



- Classification
- Localization
- Detection

- Semantic Segmentation
- Instance Segmentation

- Image Generation
- Adversarial Attacks
- etc..

Classification

Input image (C x H x W)

Cat

Output

class label (int)

Classification

Input image (C x H x W)

Cat

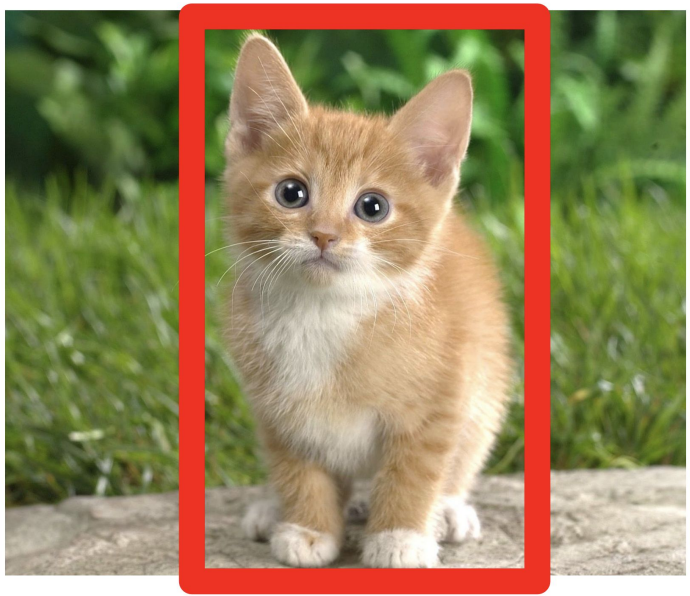
Application

- most image recognition tasks

- as a sub-task:
  - detection / localization
  - segmentation
  - prediction
  - ...

Localization

Input image (C x H x W)



Cat

Output

class label (int)

BB* coordinates (4 ints)

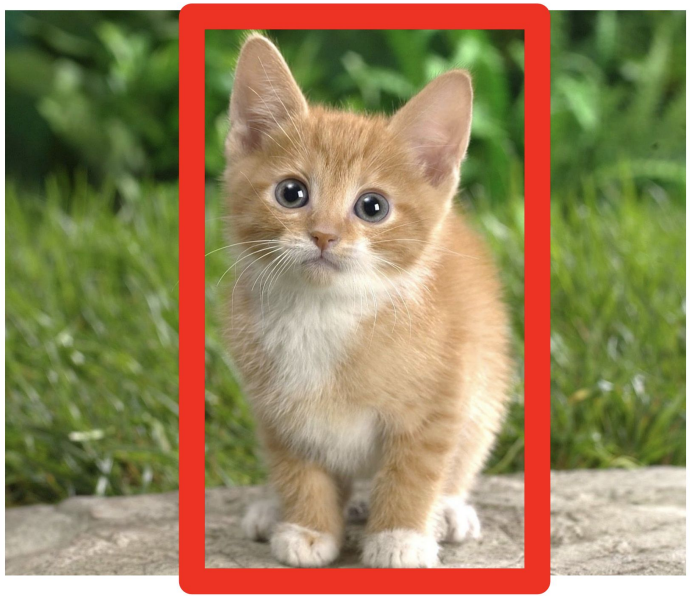
Assumption

one object per image

* BB - bounding box

Localization

Input image (C x H x W)



Cat

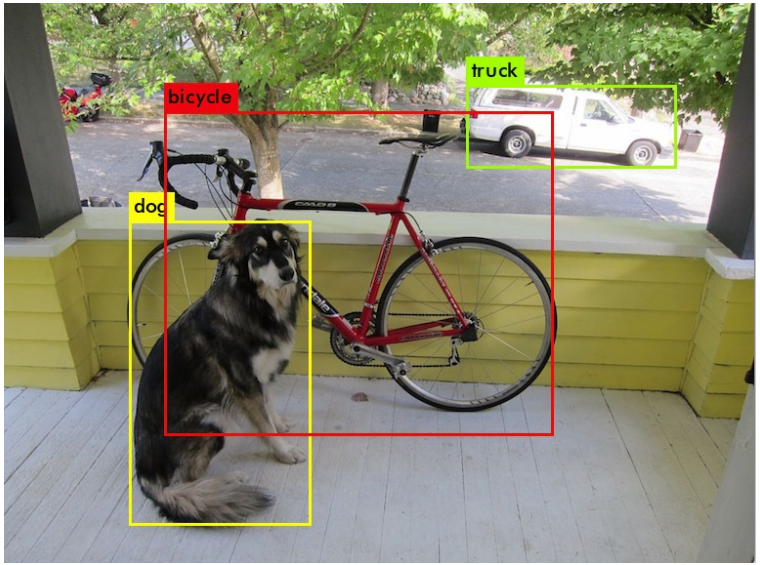
Application

- smartphone interaction
- biometrics
- gesture recognition
- robotics
- object tracking
- ...

* BB - bounding box

Detection

Input image (C x H x W)



Output

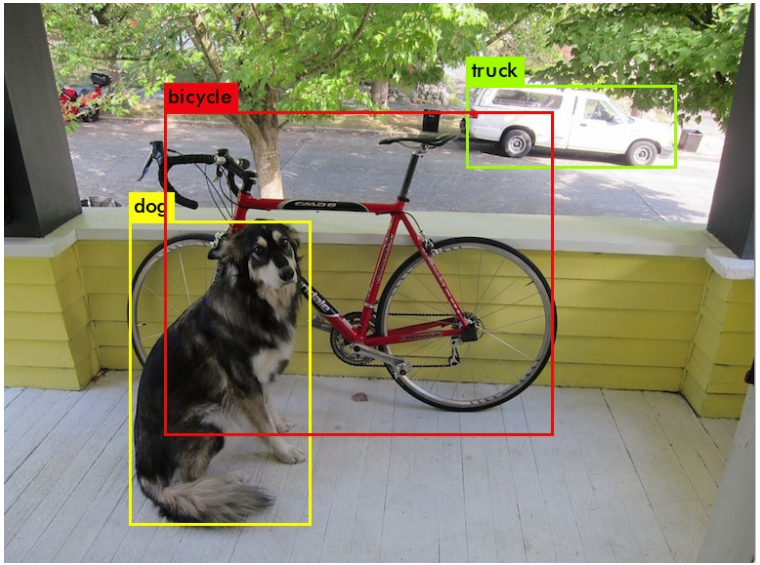
- class label (int)
- confidence (0-1 float)
- BB (4 ints)

} N times

Assumption

multiple objects per image

Input image (C x H x W)



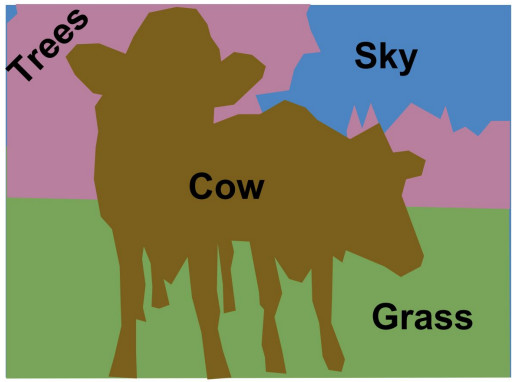
Application

- biometrics
- surveillance
- robotics
- self-driving cars
- image enhancement
- ...

Semantic Segmentation

Input image (C x H x W)

This image is CC0 public domain



Output

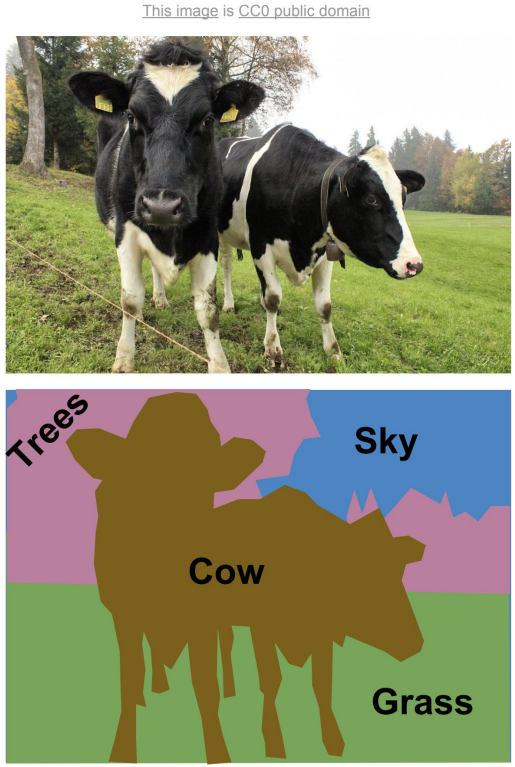
mask of classes (1 x H x W)

Assumption

just class label for each pixel

Semantic Segmentation

Input image (C x H x W)

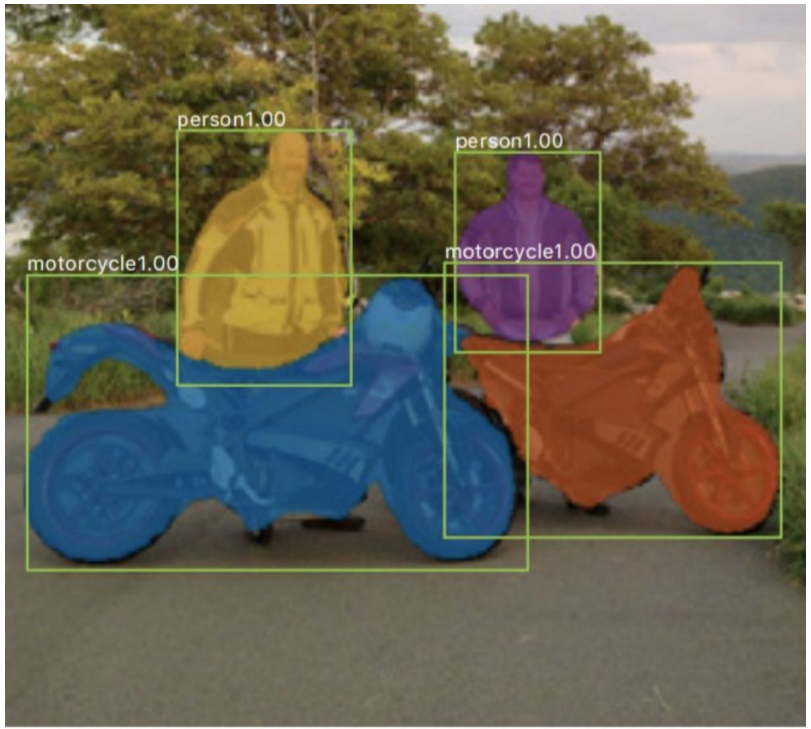


Application

- biometrics
- robotics
- self-driving cars
- image enhancement
- image super-resolution
- video recognition
- ...

Instance Segmentation

Input image (C x H x W)



Output

mask of class instances

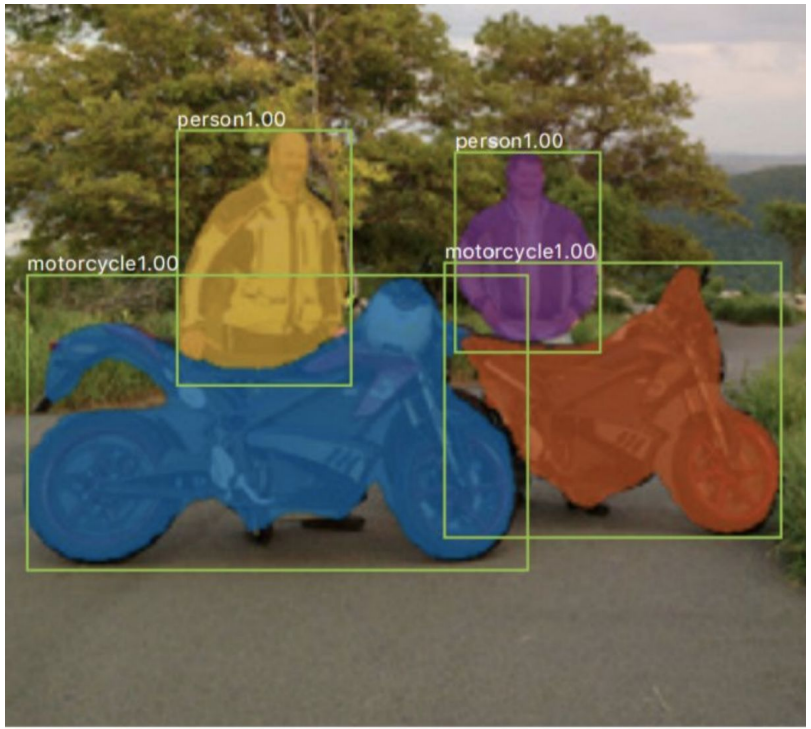
(detection + 1 x H x W)

Assumption

separated masks for each distinct object

Instance Segmentation

Input image (C x H x W)

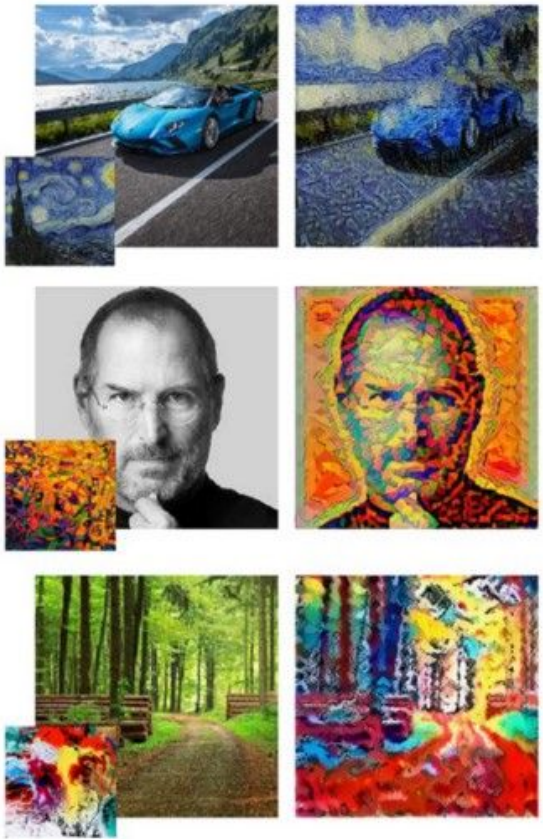


Application

- biometrics
- robotics
- self-driving cars
- image enhancement
- image super-resolution
- video recognition
- ...

Image Generation

Input image (C x H x W)



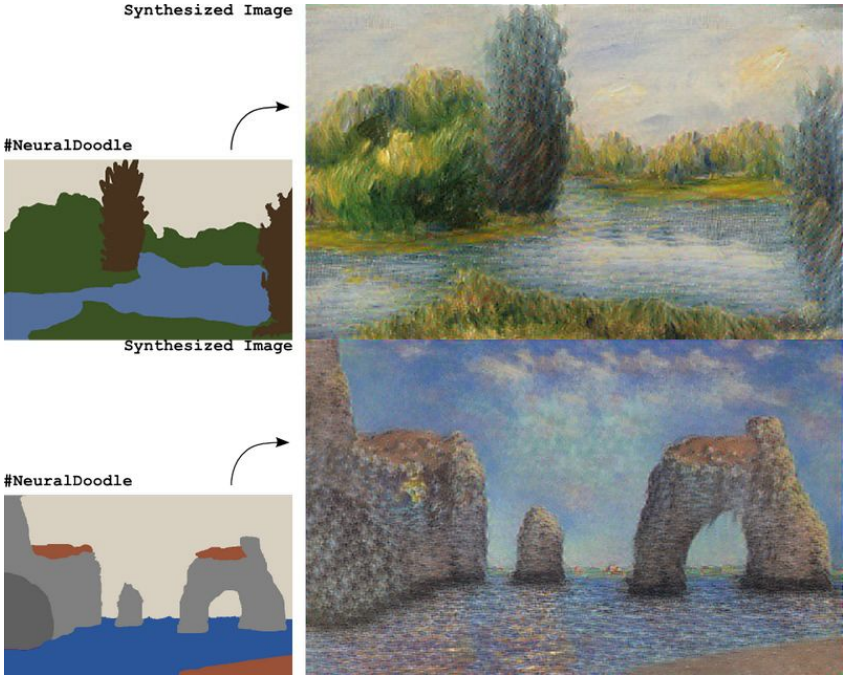
Output

image(s) of size C x H x W

image credit

Image Generation

Input image (C x H x W)



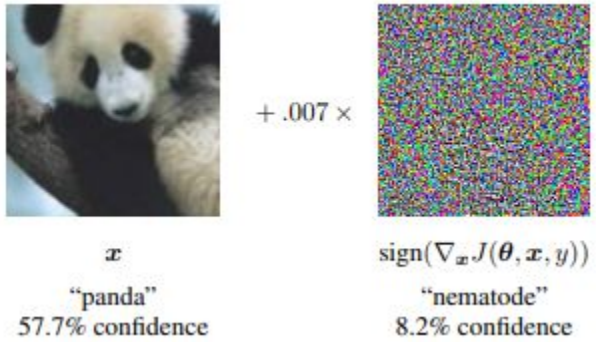
Application

- data augmentation
- dataset synthesis
- art generation
- ...

image credit

Adversarial Attacks

Input image (C x H x W)



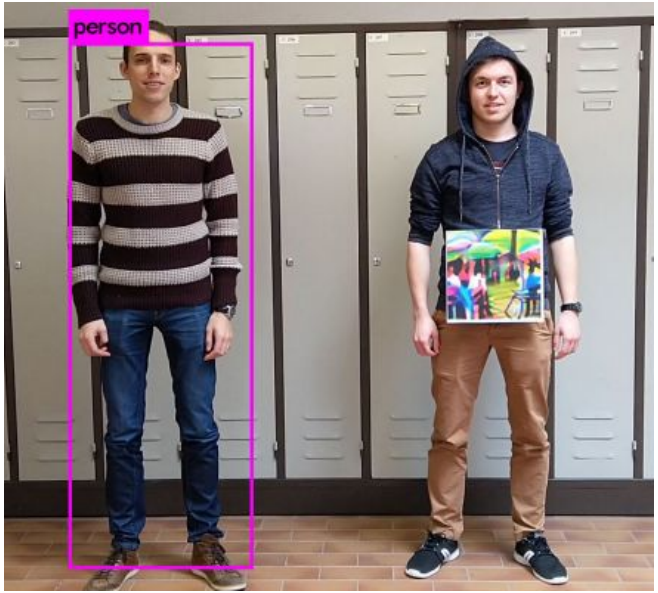
Output

image(s) of size C x H x W

image credit

Adversarial Attacks

Input image (C x H x W)



Application

- data augmentation
- model testing
- surveillance protection
- ...

https://arxiv.org/pdf/1904.08653.pdf

Computer Vision tasks: Summary

What we have discussed today:

- Classification
- Localisation
- Detection
- Semantic Segmentation
- Instance Segmentation
- Image Generation
- Adversarial Attacks

Computer Vision tasks: Summary

What we have discussed today:

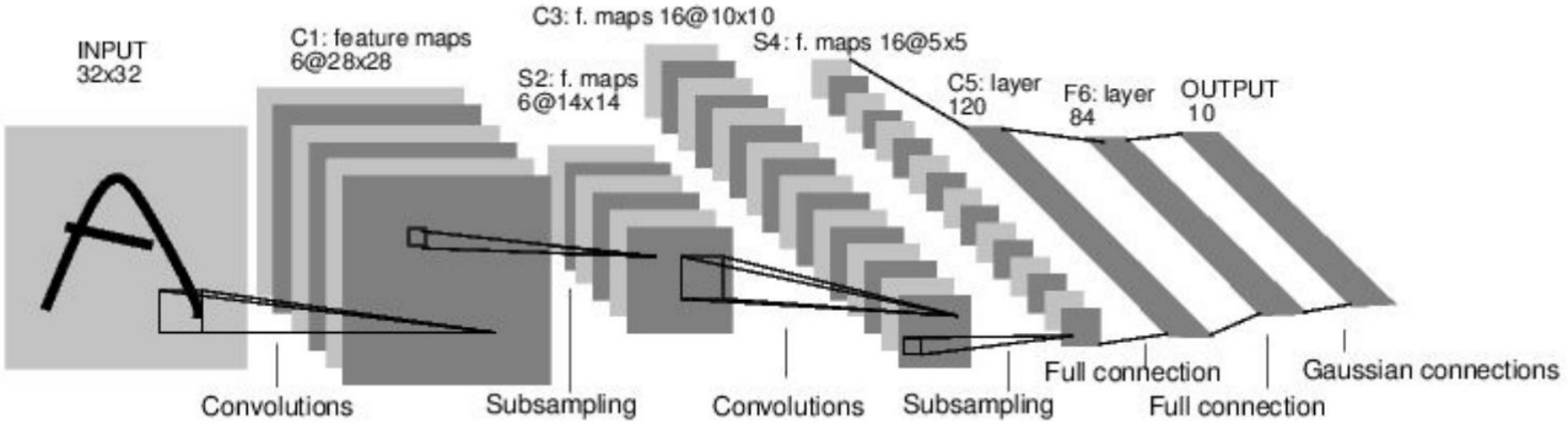
- Classification
- Localisation
- Detection
- Semantic Segmentation
- Instance Segmentation
- Image Generation
- Adversarial Attacks

What else you can do:

- Image captioning
- Image encoding
- Image morphing
- Image upscaling
- Video object tracking
- Video interpolation
- Online learning
- Scene reconstruction
- …

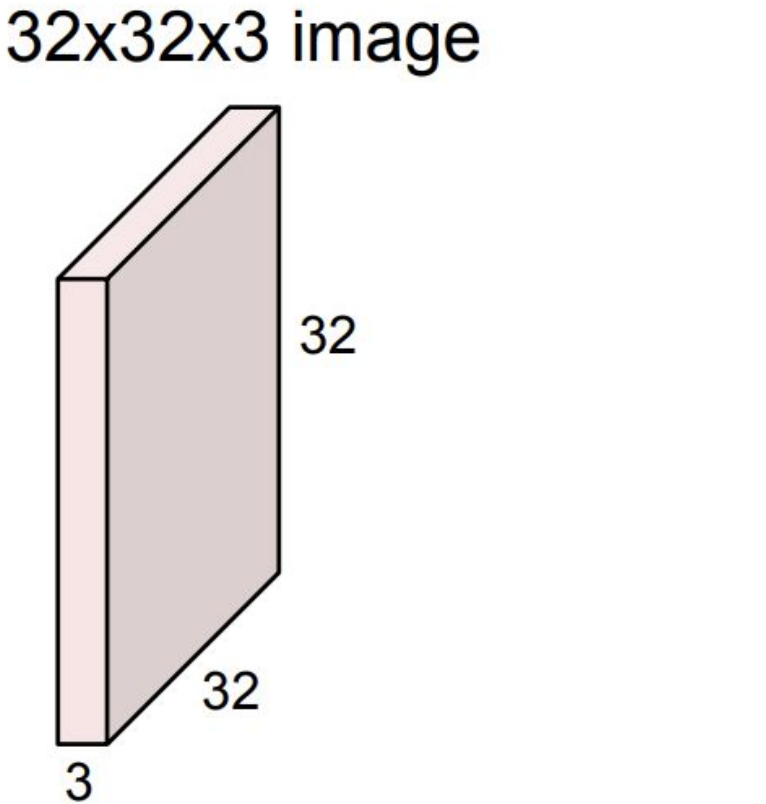
Convolutional Layer – recap

CNN



[LeNet-5, LeCun 1998]

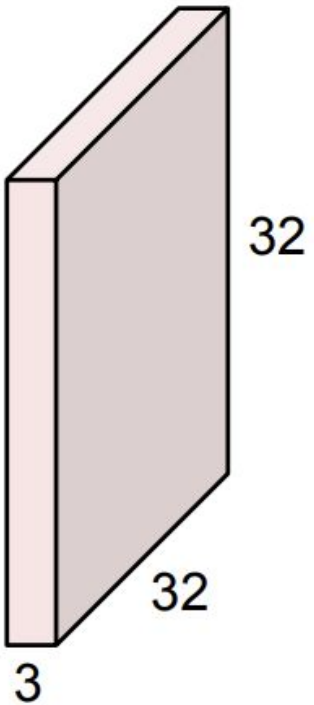
Convolutional layer



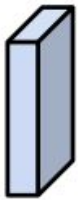
source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer

32x32x3 image



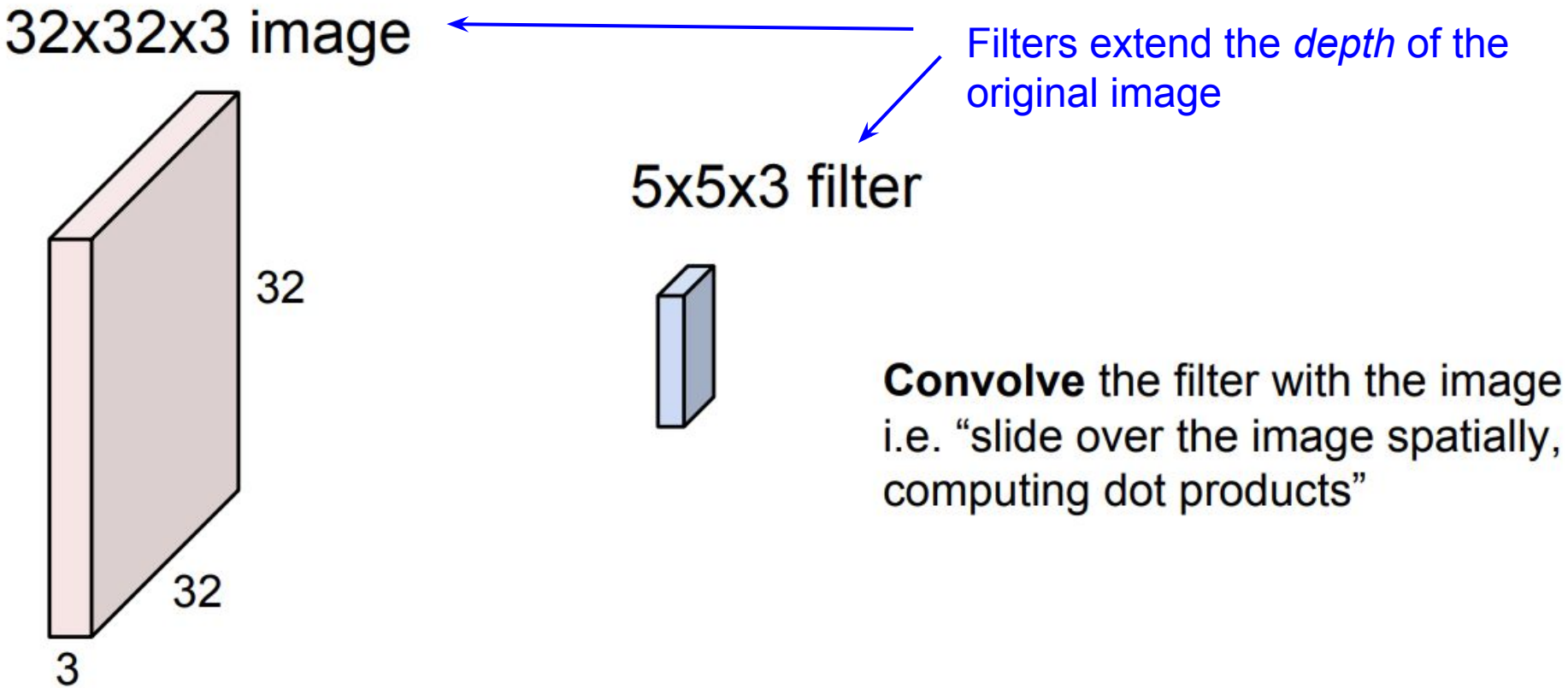
5x5x3 filter



Convolve the filter with the image
i.e. “slide over the image spatially, computing dot products”

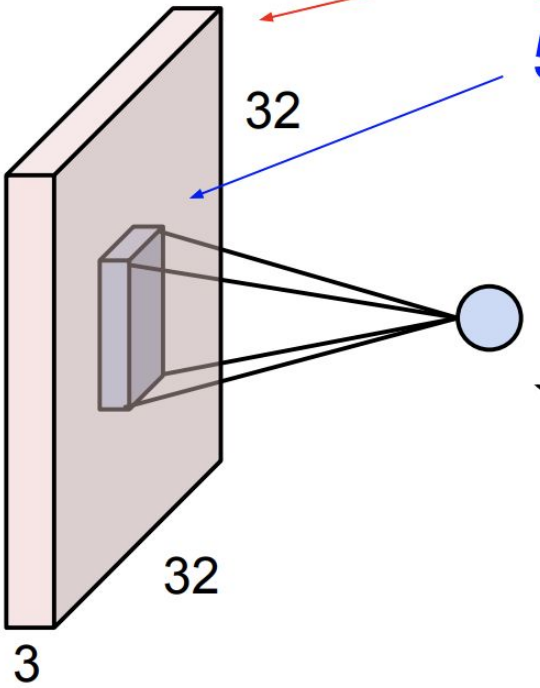
source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer



source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer



32x32x3 image

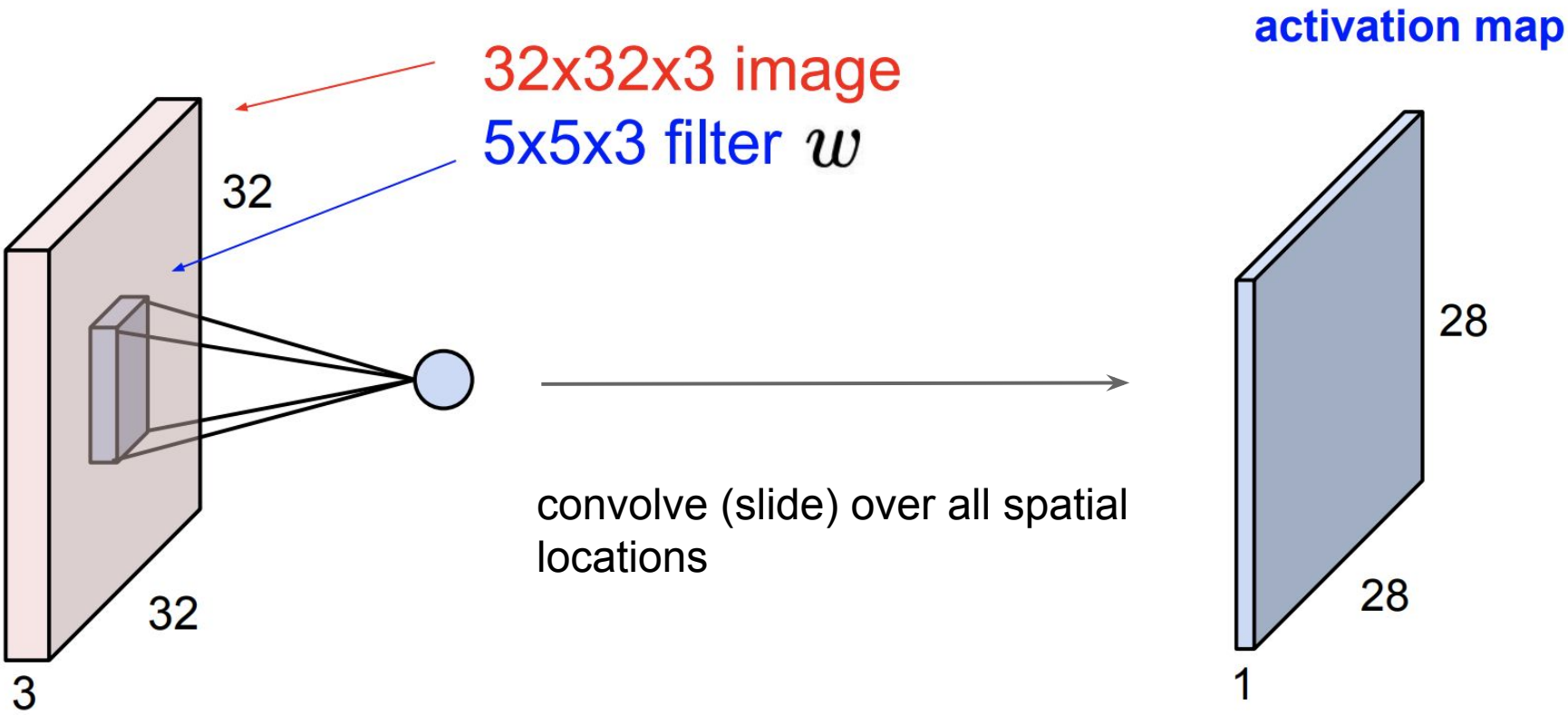
5x5x3 filter w

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer



source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer

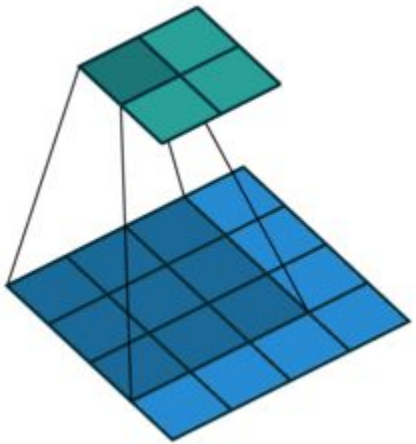
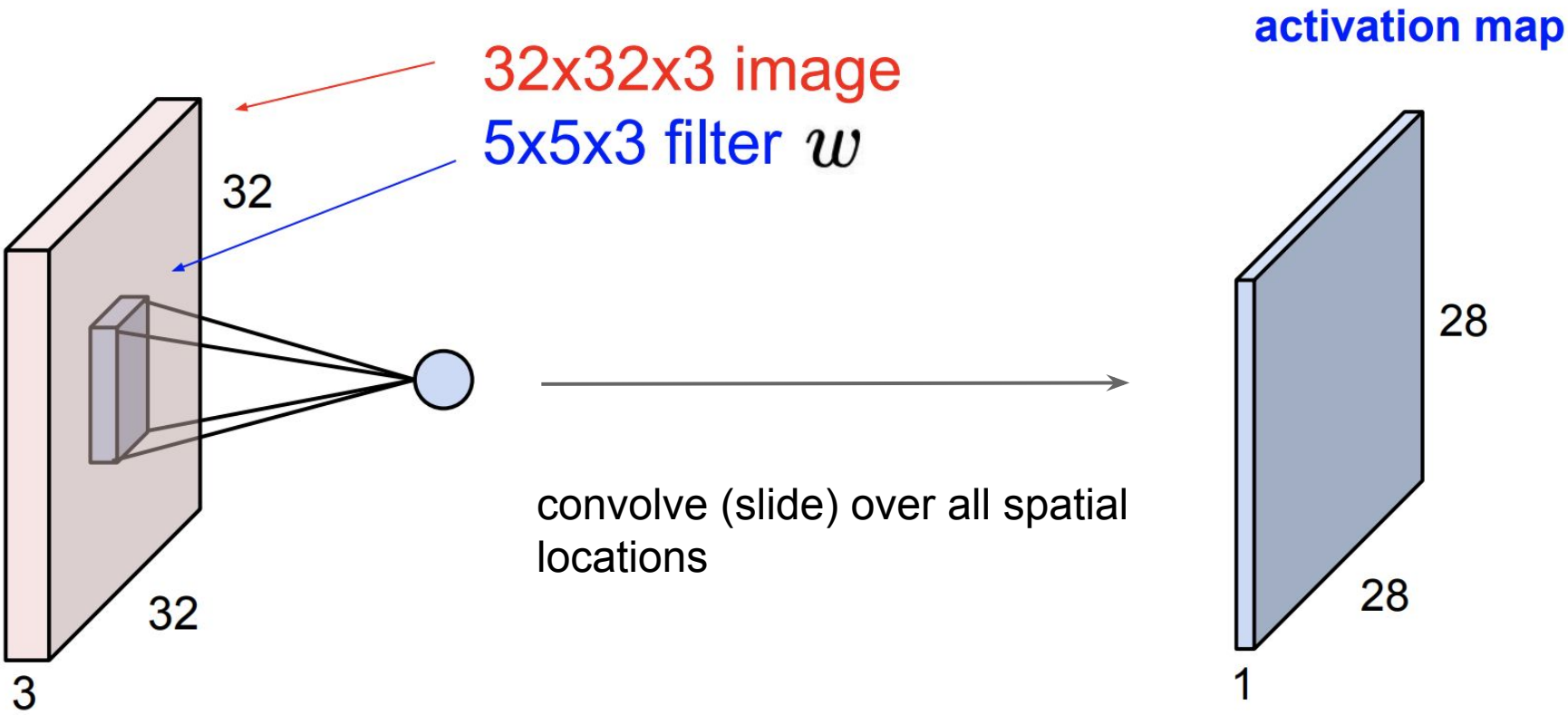


Image source: https://github.com/vdumoulin/conv_arithmetic

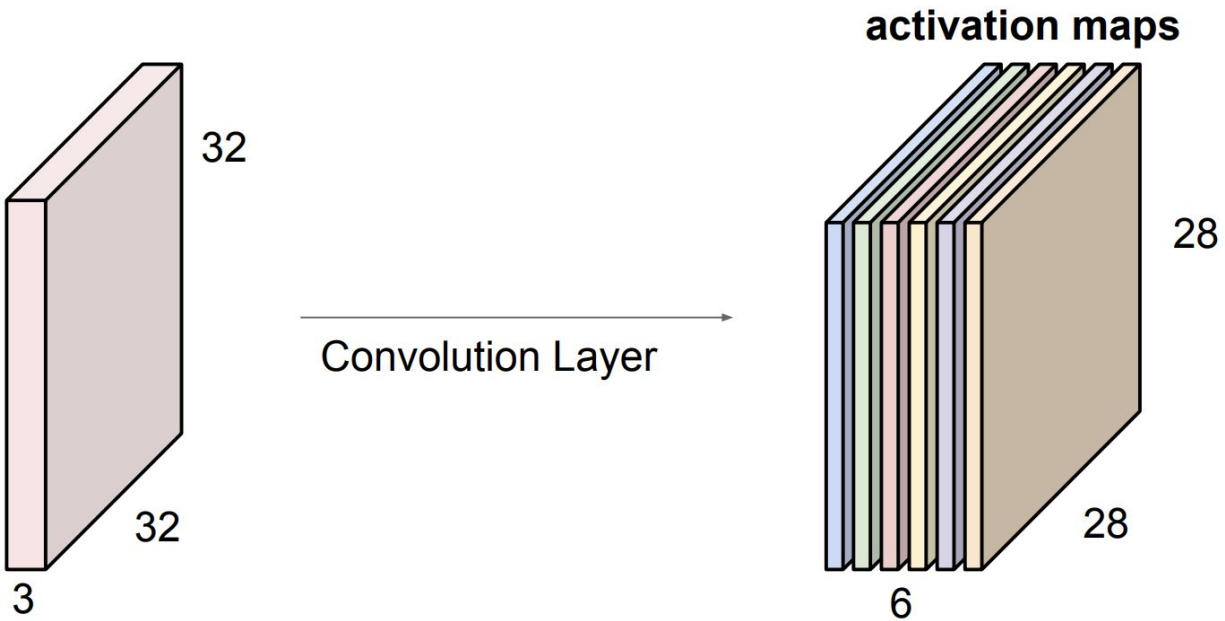
Convolutional layer



source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

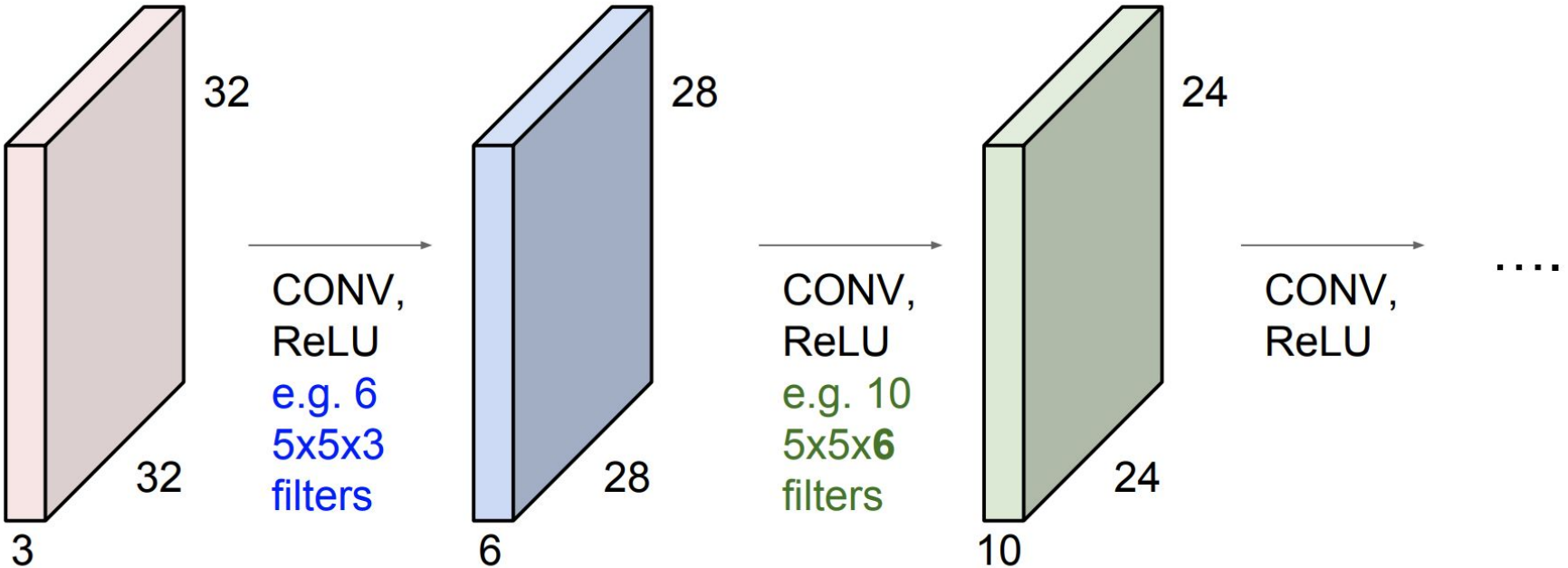


We stack these up to get a "new image" of size 28x28x6!

source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Convolutional layer

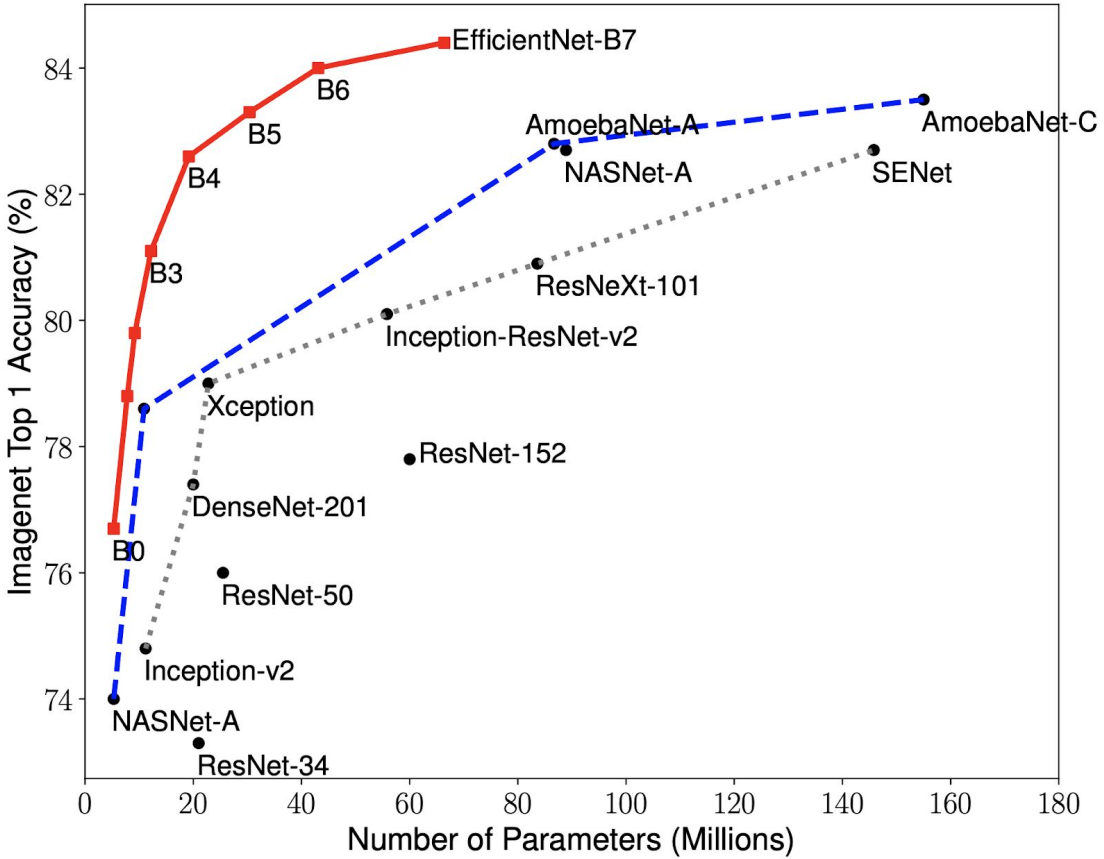
Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



source: http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

Famous NN Architectures in Computer Vision

Overview



- Usually Deeper networks show better results

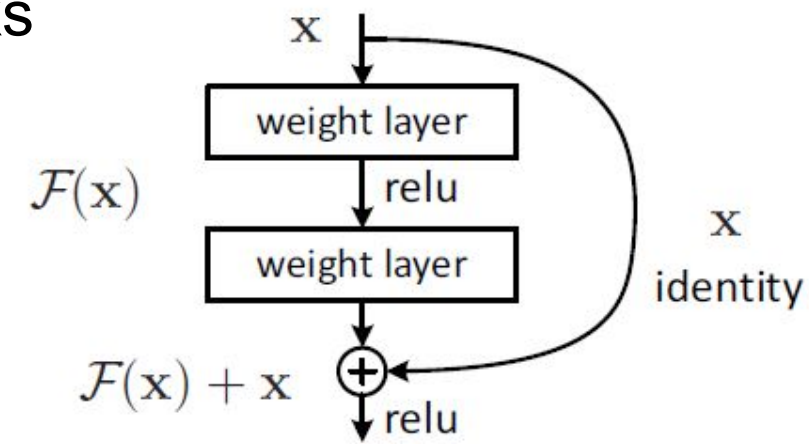
- But there are problems:
  - vanishing gradients
  - exploding gradients
  - biased activations
  - memory
  - speed
  - overfitting

image credit

ResNet [2015]

Main feature – residual blocks:

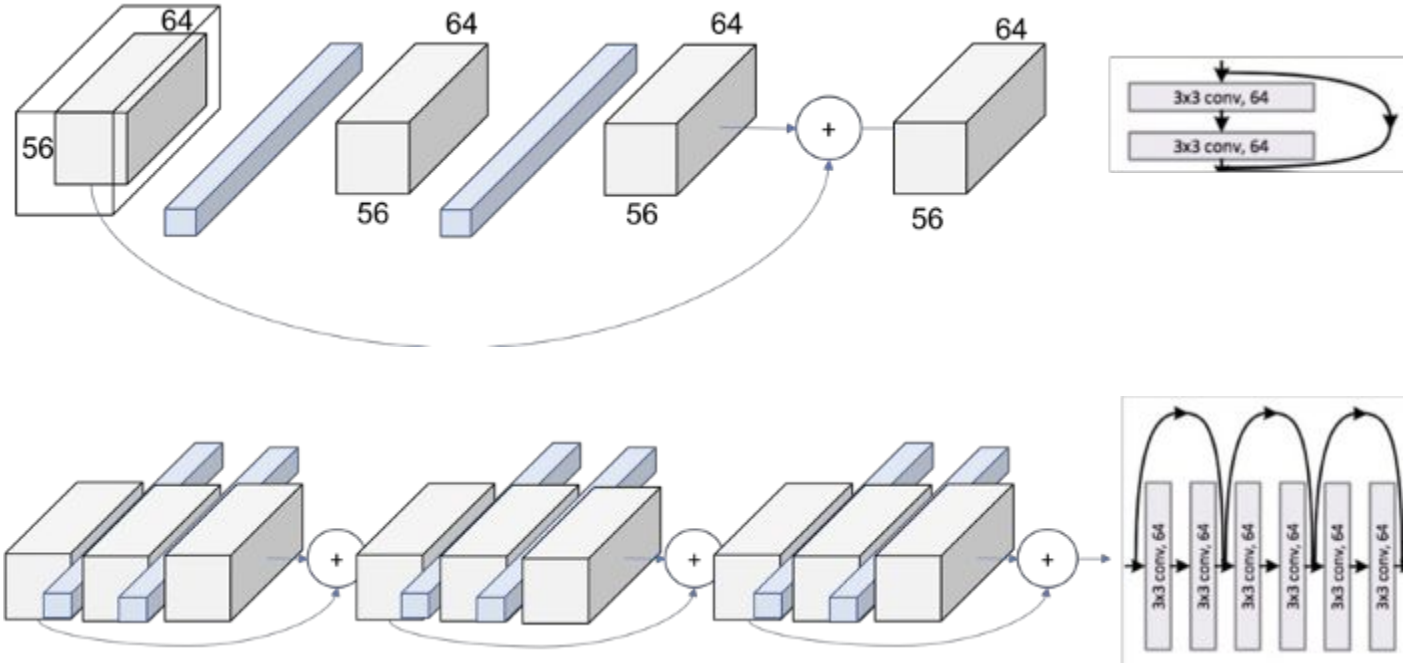
- Allow training deeper networks
- Improve vanishing gradient problem
- Look like LSTM



A Building Block of Residual Network

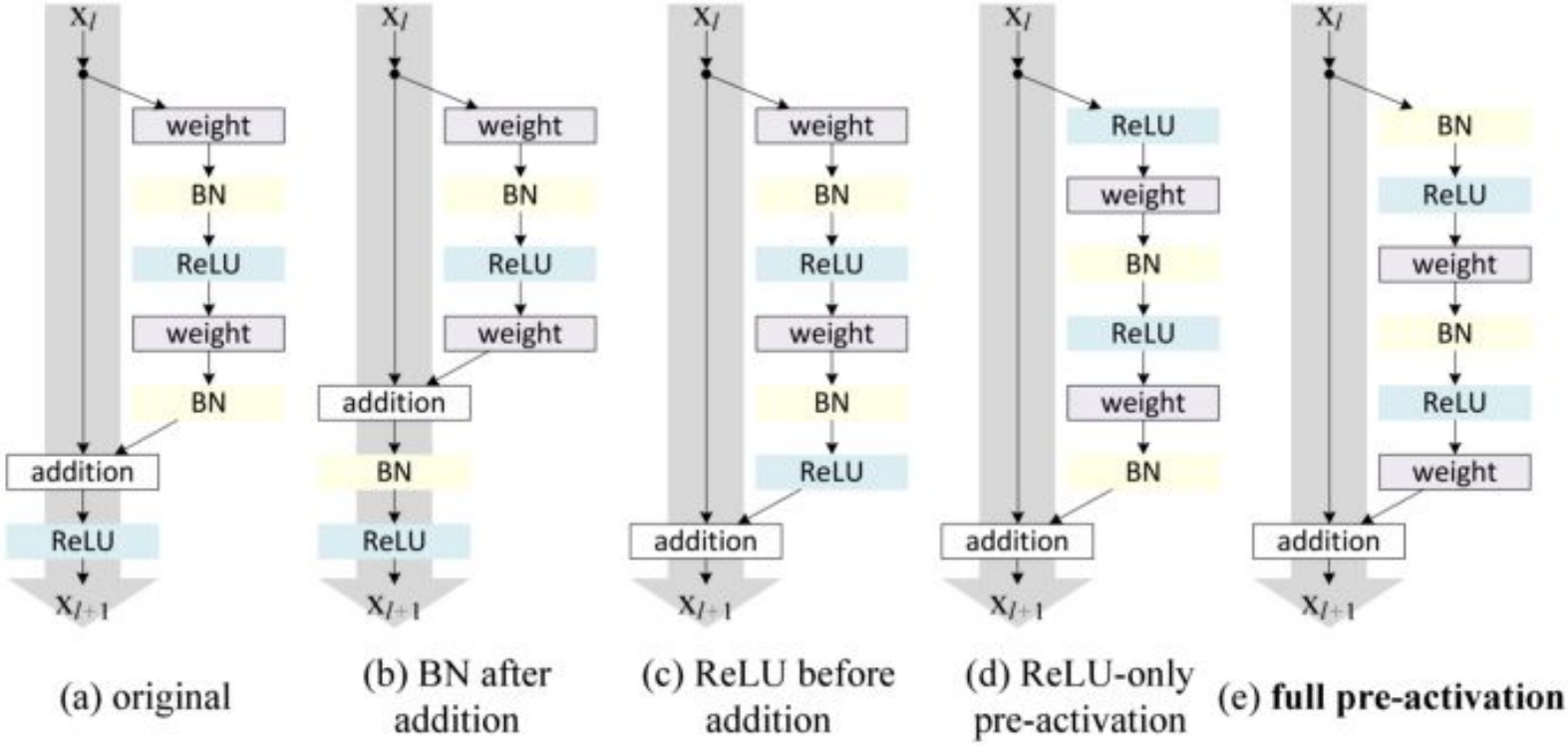
https://arxiv.org/pdf/1512.03385.pdf

ResNet [2015]



https://arxiv.org/pdf/1512.03385.pdf

ResNet [2015]

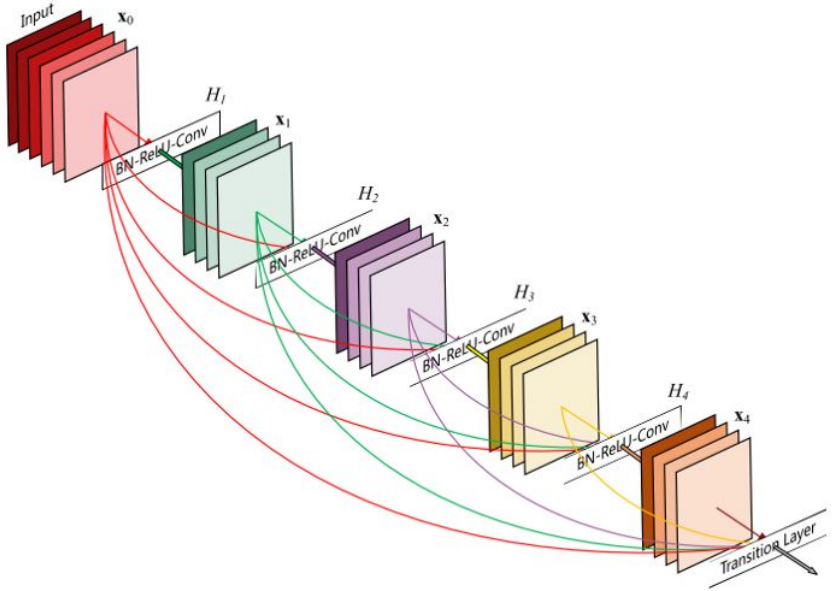


https://arxiv.org/pdf/1603.05027.pdf

DenseNet [2016]

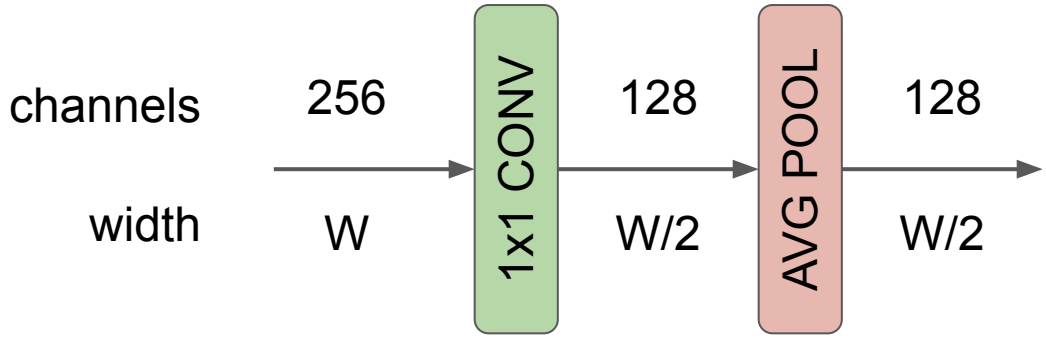
Features:

- Connects each layer to every other layer in a feed-forward fashion
- Concatenate outputs
- Improves vanishing gradient problem



https://arxiv.org/pdf/1608.06993v3.pdf

DenseNet [2016]



Transition block visualization

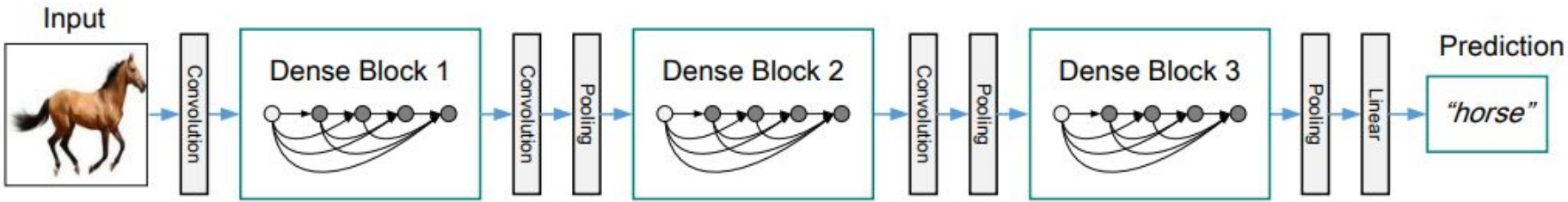
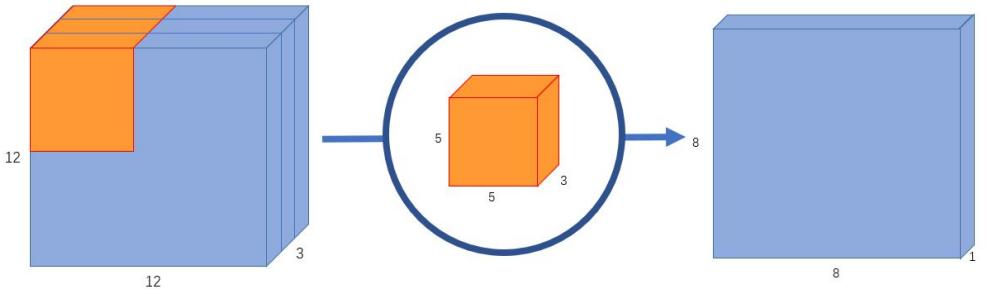


Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

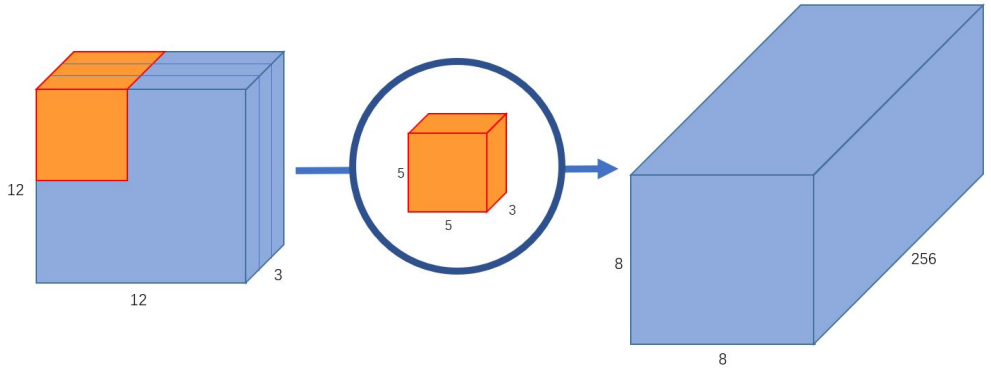
https://arxiv.org/pdf/1608.06993v3.pdf

Normal convolution

- Uses all input channels:
- Is it redundant?



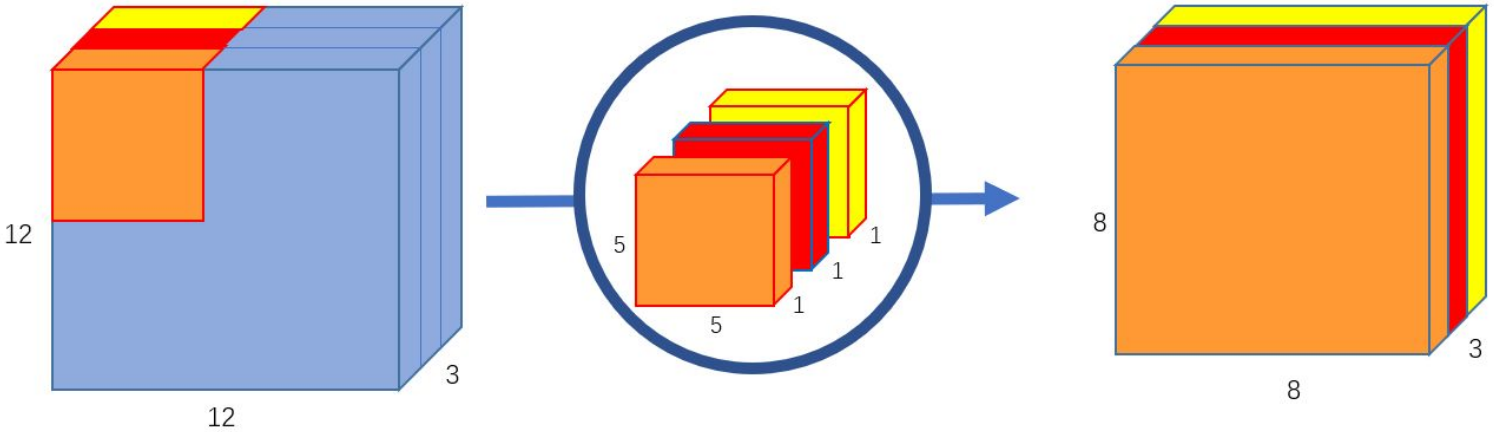
- Kernel may be factorized
  - channels: **depthwise**
  - spatially: **separable**



→ Less memory / FLOPs

Kernel size: C x K x K x H

Convolutions: Depthwise

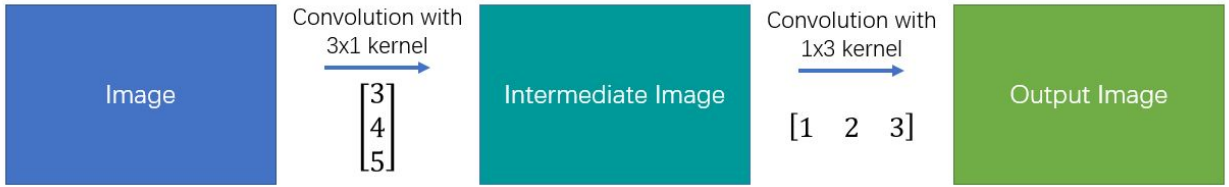


Keep channel number

Kernel size: C x K x K

Convolutions: Separable

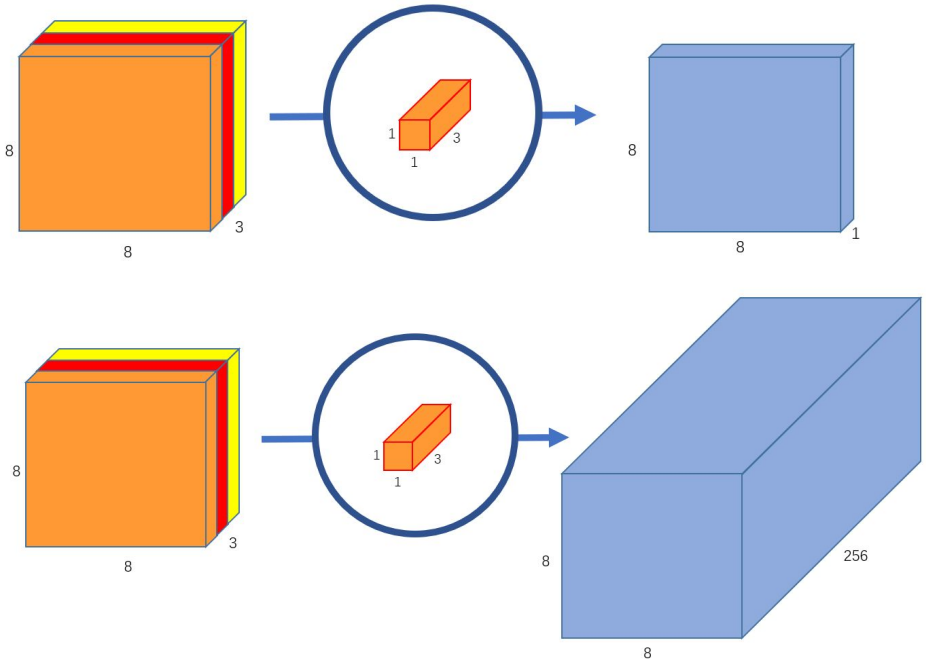
$$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$



Keep channel number

Filter size: C x (K + K)

Convolutions: Pointwise



Change only channel number

Filter size: C x 1 x 1 x H

Convolutions

Compare by kernel size:

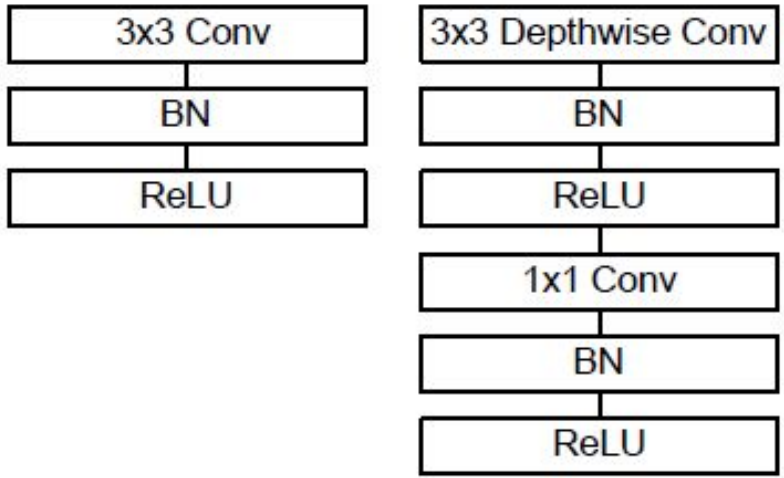
- Normal: C x K x K x H
- Depthwise: C x K x K → **H** times smaller
- Separable: C x (K + K) → **0.5 x K x H** times smaller
- Pointwise: C x 1 x 1 x H → **K x K** times smaller

Can combine them!

MobileNet [2017]

Features:

- Split convolution
  - “depthwise” part
  - “pointwise” part
- Less parameters
- Same performance
- Use in low-power devices

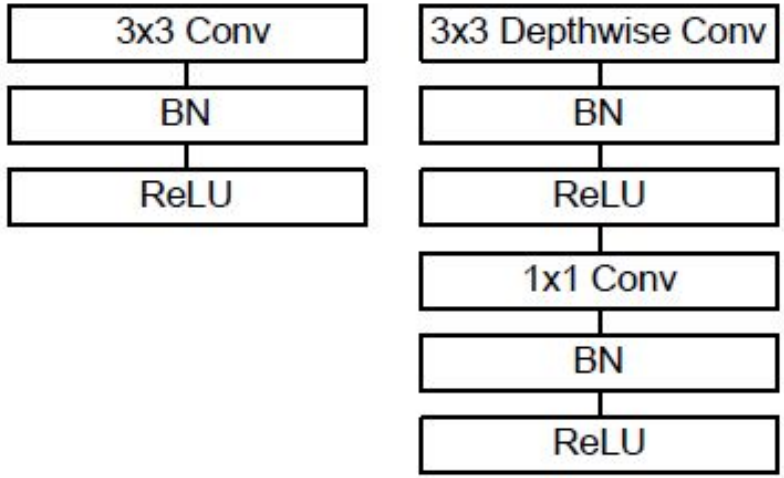


https://arxiv.org/abs/1704.04861

MobileNet [2017]

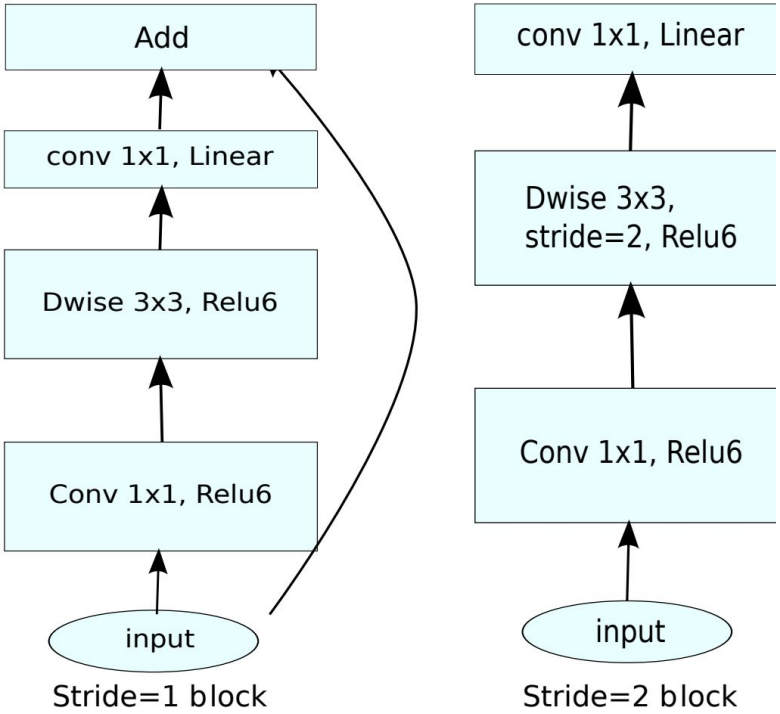
Table 1. MobileNet Body Architecture

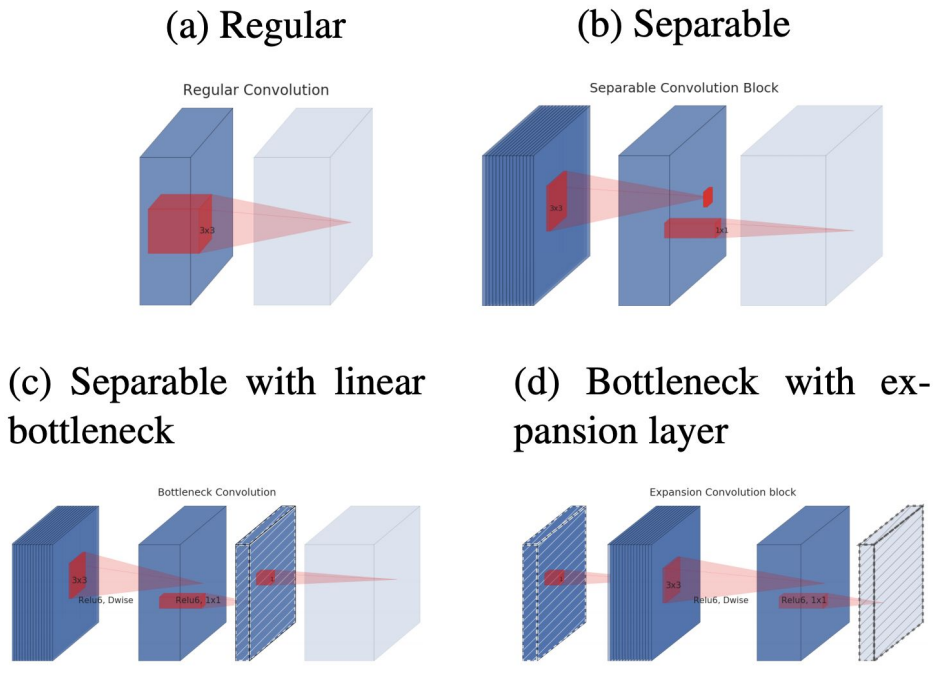
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$



https://arxiv.org/abs/1704.04861

MobileNet v2 [2018]





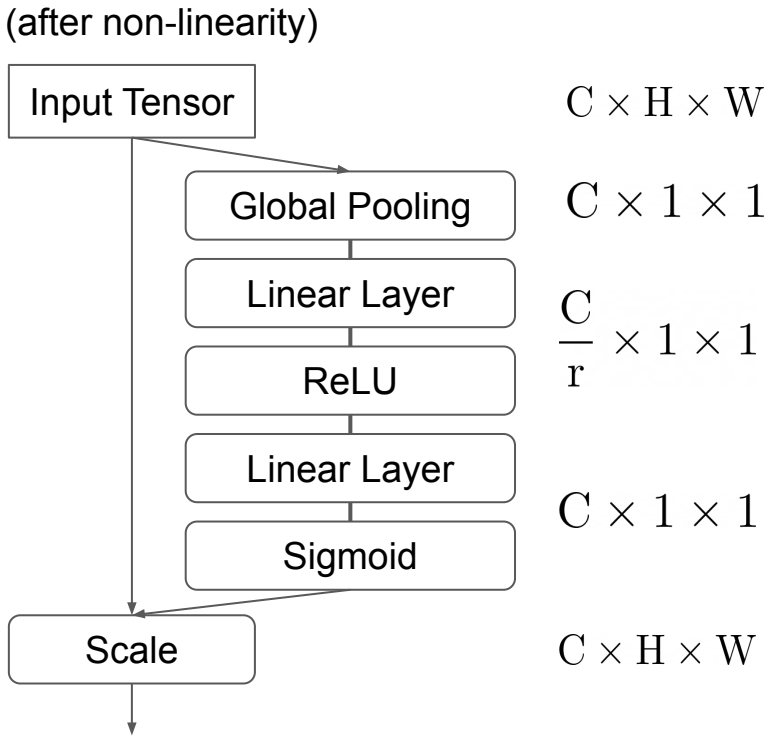
https://arxiv.org/abs/1801.04381

Squeeze & Excitation Networks [2018]

Features:

- Use global averages of feature maps for better classification
- Scale feature map channels
- “Excite” useful info
- Suppress redundant info
- Apply: deep CNNs’ later layers

Basic SE Block



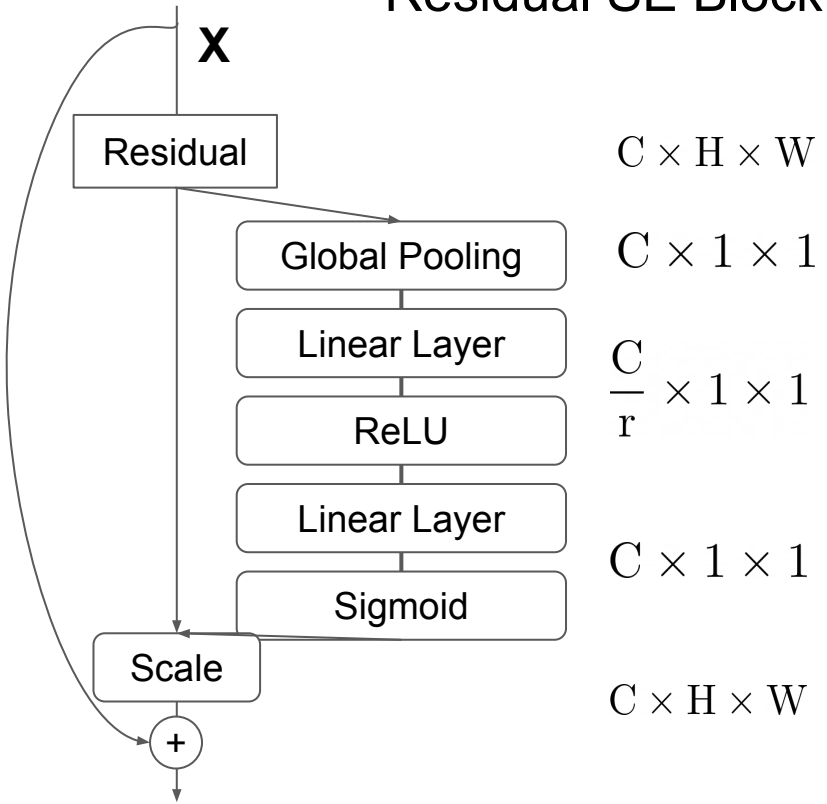
https://arxiv.org/pdf/1709.01507.pdf

Squeeze & Excitation Networks [2018]

Features:

- Use global averages of feature maps for better classification
- Scale feature map channels
- “Excite” useful info
- Suppress redundant info
- Apply: deep CNNs’ later layers

Residual SE Block



https://arxiv.org/pdf/1709.01507.pdf

EfficientNet [2018]

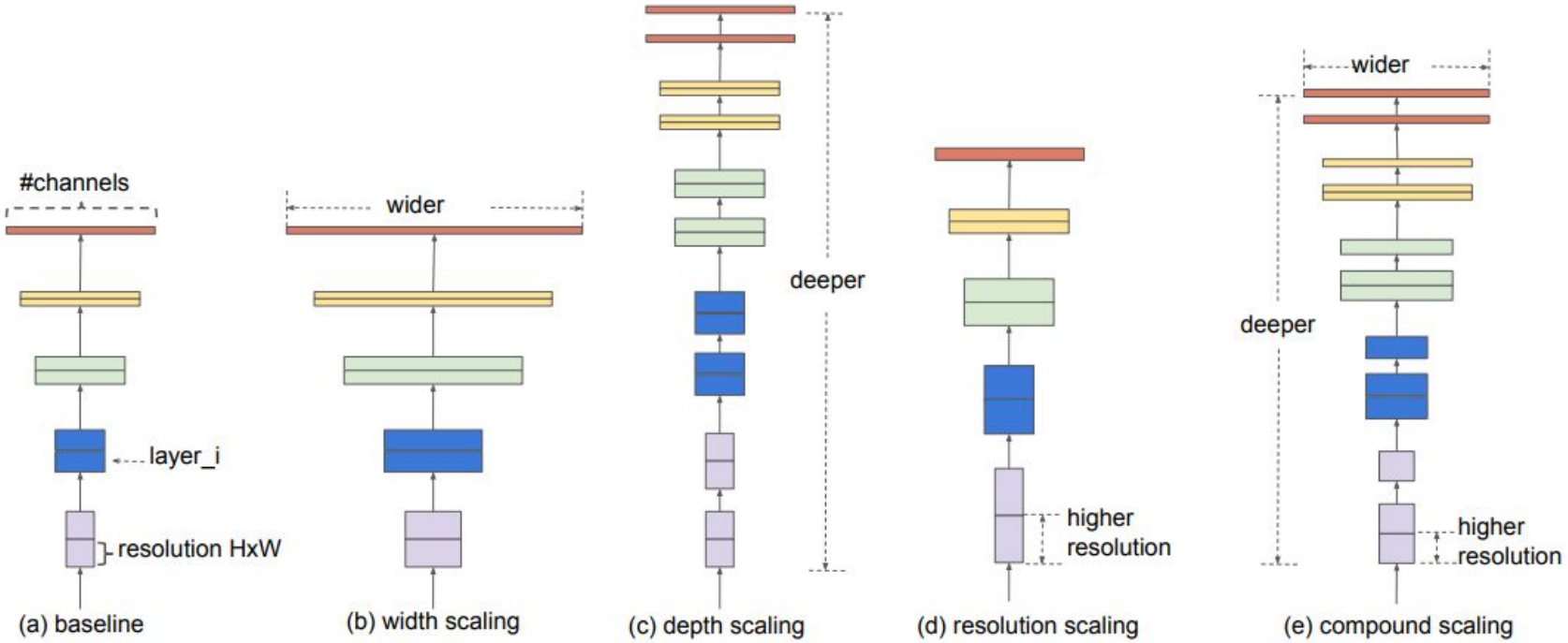


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

https://arxiv.org/pdf/1905.11946.pdf

EfficientNet [2018]

Features:

- Scaling the model by:
  - depth (layers)
  - width (channels)
  - input resolution
- Compound scaling for better performance
- Neural architecture search

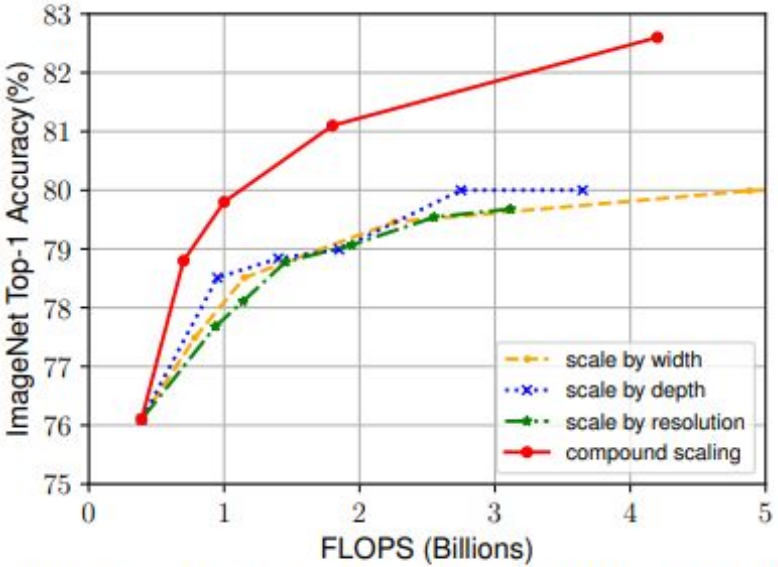


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

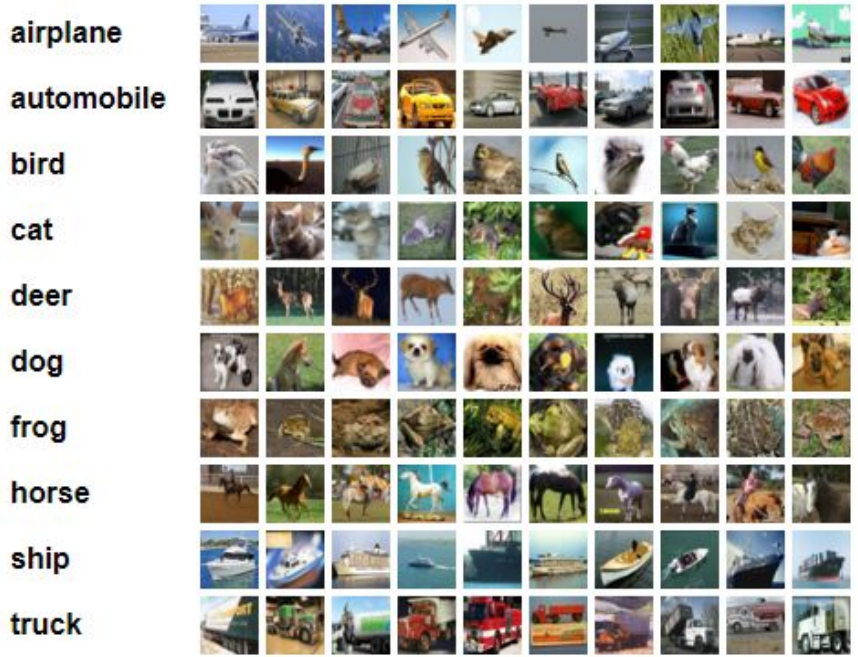
https://arxiv.org/pdf/1905.11946.pdf

Common Datasets

CIFAR10 & CIFAR100

- 60K 32x32 colour images
- 10 classes
- 6K images per class

  or
- 100 classes
- 600 images per class



http://www.cs.toronto.edu/~kriz/cifar.html

PASCAL VOC 2012

- (Visual Object Classes)
- developed since 2005
- 11,5k images
- 20 classes
- 31,5k detections
- 7k segmentations
- Images came from flickr.com

Table 1 The VOC classes

Vehicles	Household	Animals	Other
Aeroplane	Bottle	Bird	Person
Bicycle	Chair	Cat	
Boat	Dining table	Cow	
Bus	Potted plant	Dog	
Car	Sofa	Horse	
Motorbike	TV/Monitor	Sheep	
Train			

The classes can be considered in a notional taxonomy

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

http://host.robots.ox.ac.uk/pascal/VOC/pubs/everingham15.pdf

Datasets

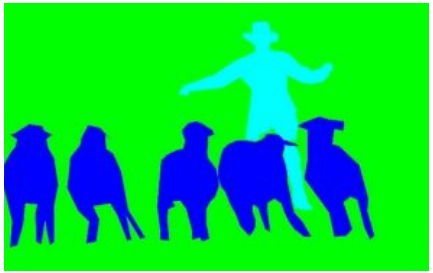
ImageNet

- WordNet hierarchy
- 14.2m images
- 1000 classes
- 1m images with BBs
- diverse objects properties

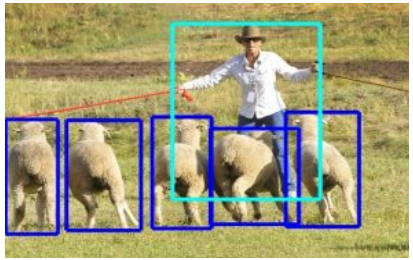
http://image-net.org/explore

MS COCO 2017

- 328k images
- 91 object categories
- 82 with more than 5,000 labeled instances
- 2,5m labeled instances
- 250k people with keypoints


semantic segmentation

classification


object localization


instance segmentation

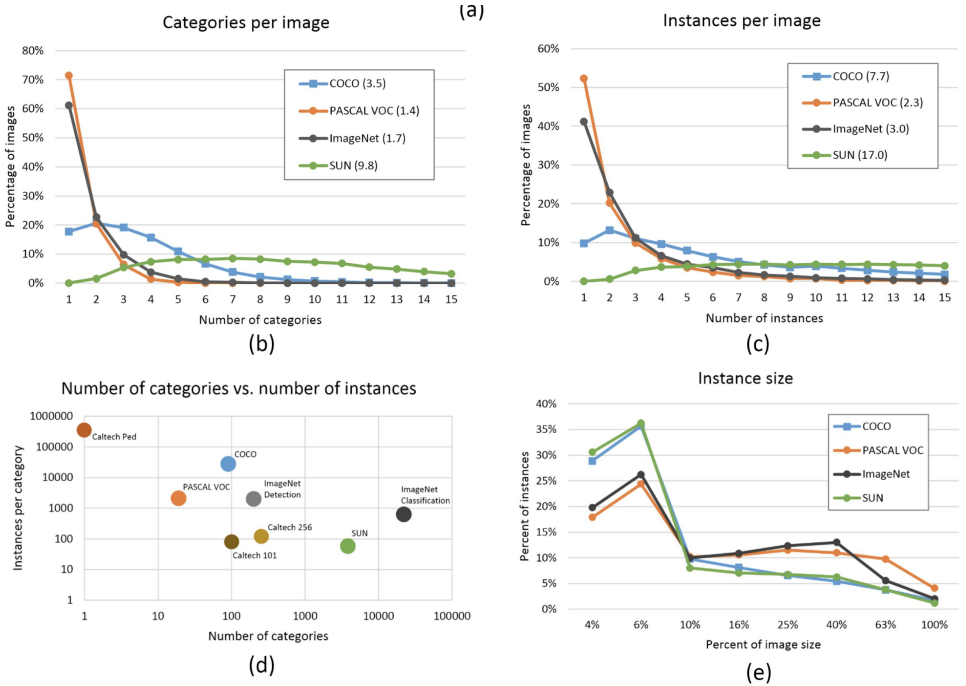
https://arxiv.org/pdf/1405.0312.pdf

http://cocodataset.org/#explore

Datasets

MS COCO 2017

- 328k images
- 91 object categories
- 82 with more than 5,000 labeled instances
- 2,5m labeled instances
- 250k people with keypoints

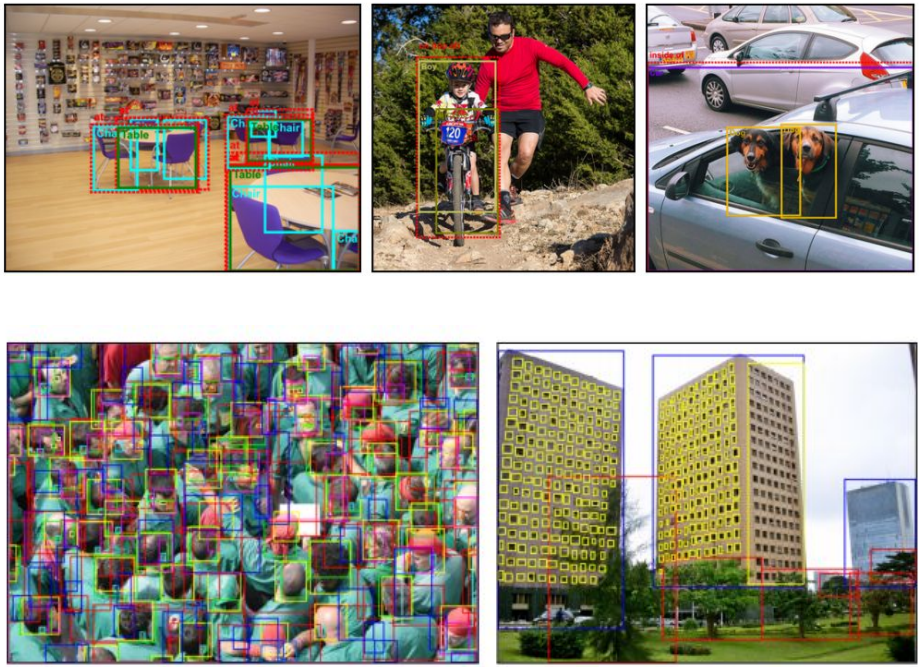


https://arxiv.org/pdf/1405.0312.pdf

http://cocodataset.org/#explore

Open Images

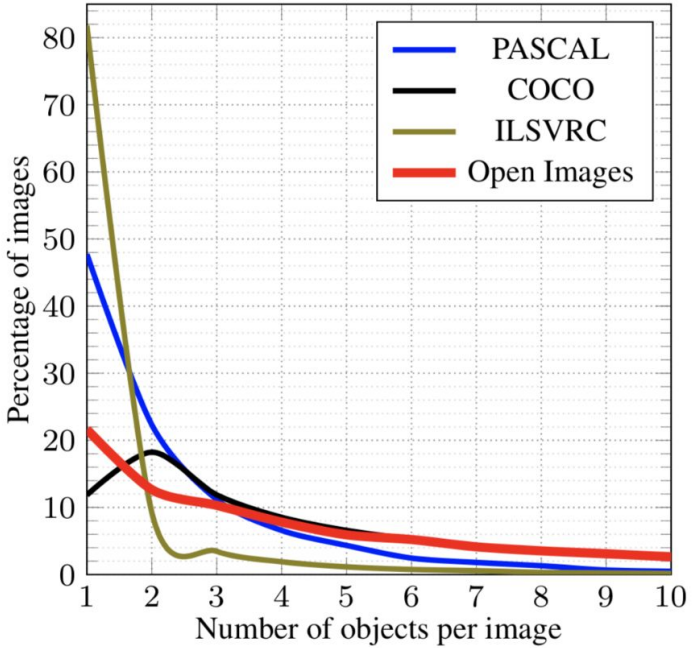
- Currently relevant challenge
- 15m boxes on 600 categories
- 2,8m instance segmentations on 350 categories
- 36,5m image-level labels on 20k categories
- 391k relationship annotations of 329 relationships



https://storage.googleapis.com/openimages/web/factsfigures.html

Open Images

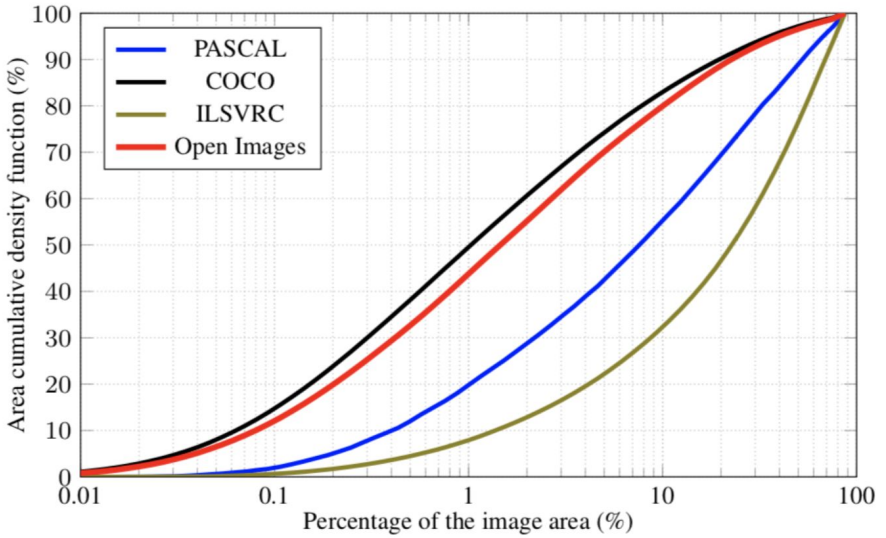
- Currently relevant challenge
- 15m boxes on 600 categories
- 2,8m instance segmentations on 350 categories
- 36,5m image-level labels on 20k categories
- 391k relationship annotations of 329 relationships



https://storage.googleapis.com/openimages/web/factsfigures.html

Open Images

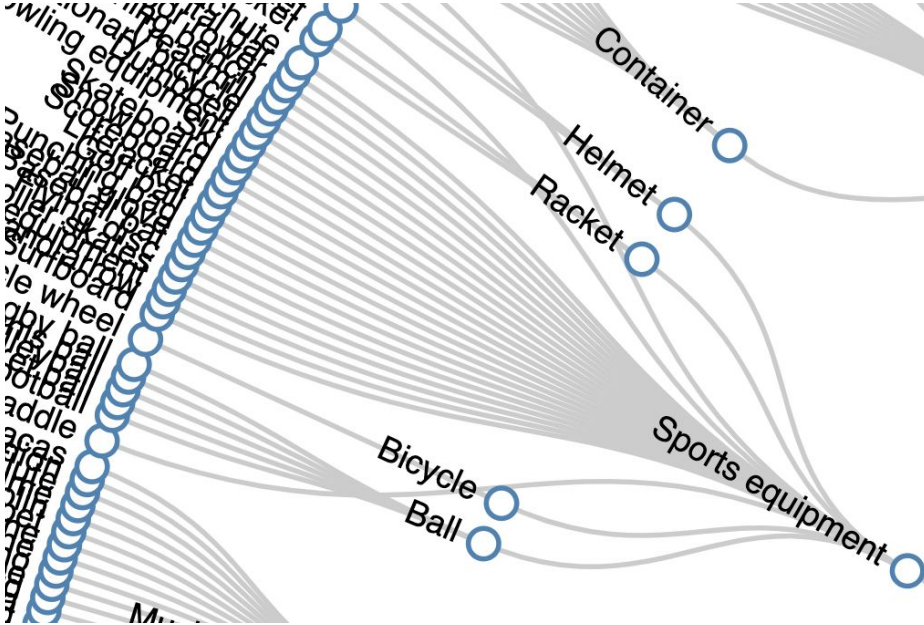
- Currently relevant challenge
- 15m boxes on 600 categories
- 2,8m instance segmentations on 350 categories
- 36,5m image-level labels on 20k categories
- 391k relationship annotations of 329 relationships



https://storage.googleapis.com/openimages/web/factsfigures.html

Open Images

- Currently relevant challenge
- 15m boxes on 600 categories
- 2,8m instance segmentations on 350 categories
- 36,5m image-level labels on 20k categories
- 391k relationship annotations of 329 relationships



https://storage.googleapis.com/openimages/web/factsfigures.html

Summary

- Vanishing and exploding gradients are still present
- Careful model scaling is important
- Convolution layers can be factorized
- Datasets are getting better
- A lot of Computer Vision tasks yet to be solved