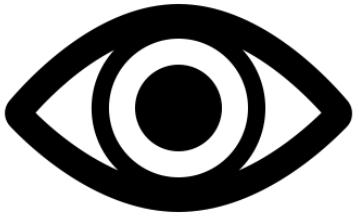


Advanced Image Processing
Exploring Architectures

SYDE 671

Big space of designs!

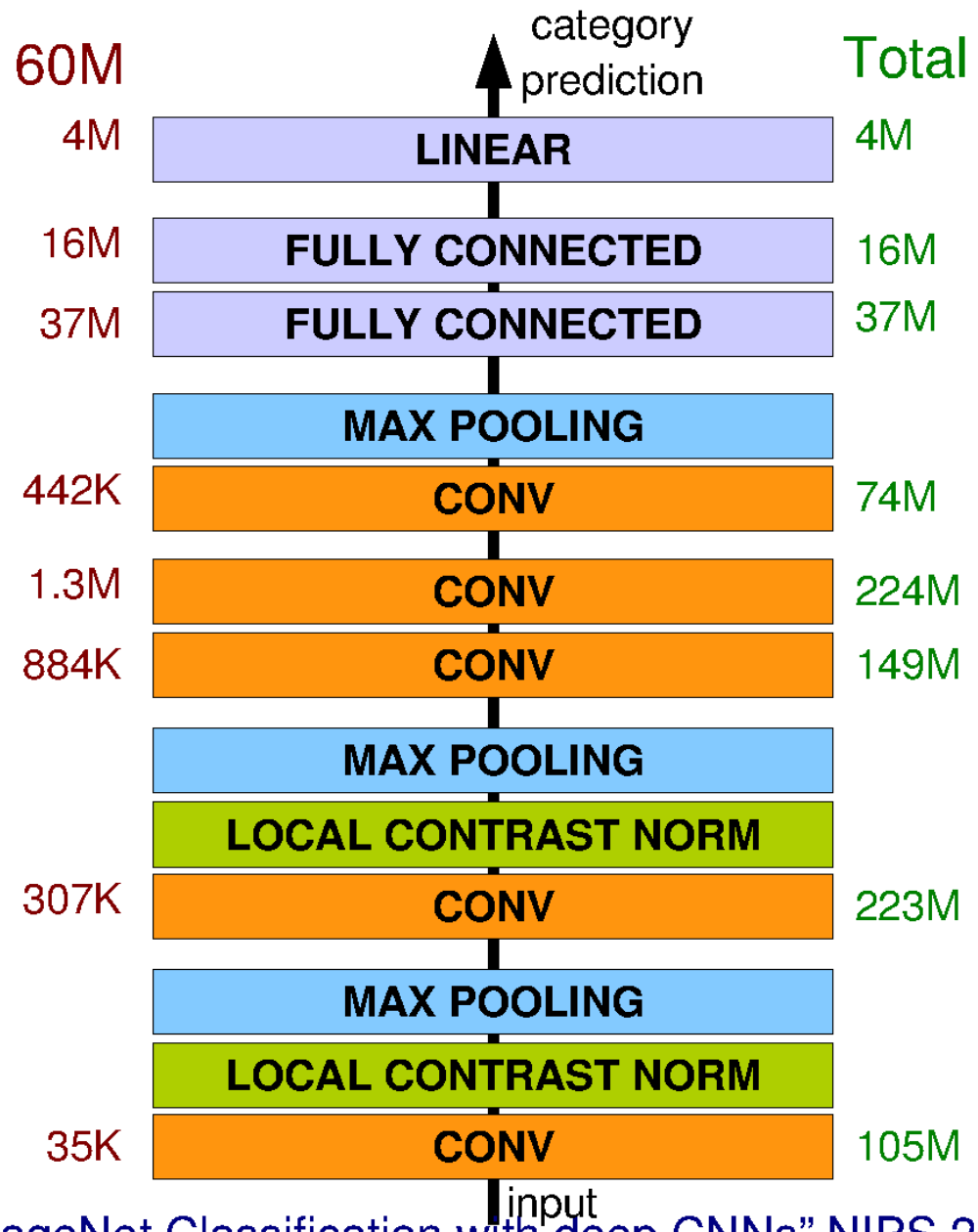
We still don’t even know how many layers we need.

Deep-CNN

AlexNet

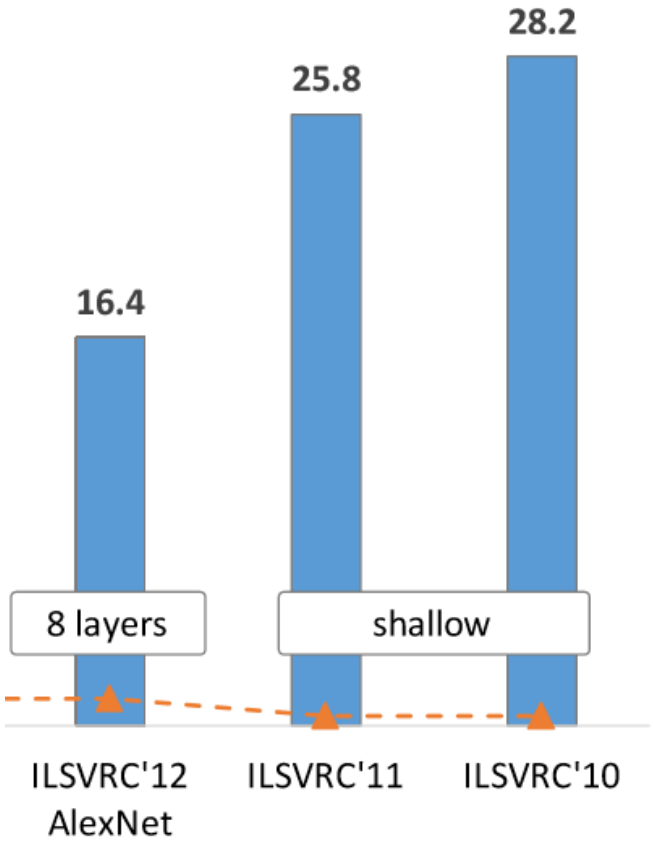
Total nr. params: 60M

Total nr. flops: 832M



Krizhevsky et al. "ImageNet Classification with deep CNNs" NIPS 2012

Revolution of Depth



ImageNet Classification top-5 error (%)


WE NEED TO GO
DEEPER

Beyond AlexNet

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan & Andrew Zisserman 2015

These are the pre-trained "VGG" networks that you use in assignment 4

VGG = Visual Geometry Group at Oxford, UK

ConvNet Configuration					
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 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

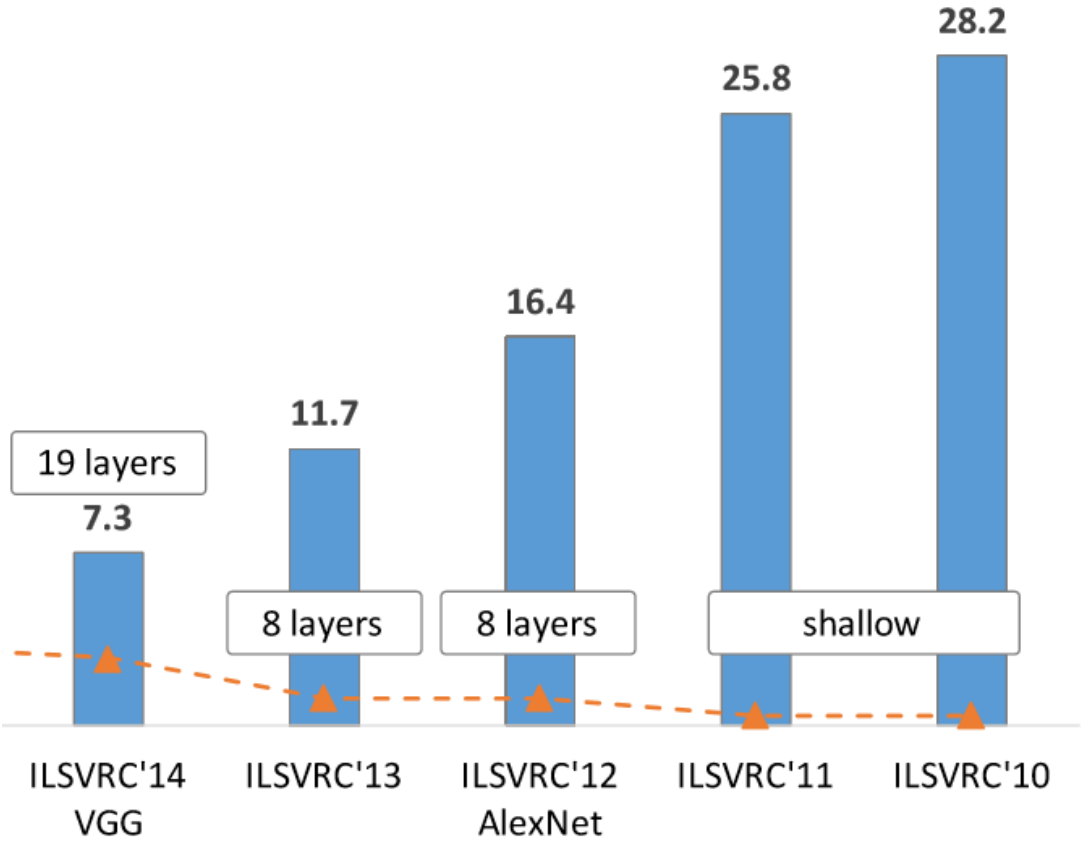
Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Table 4: **ConvNet performance at multiple test scales.**

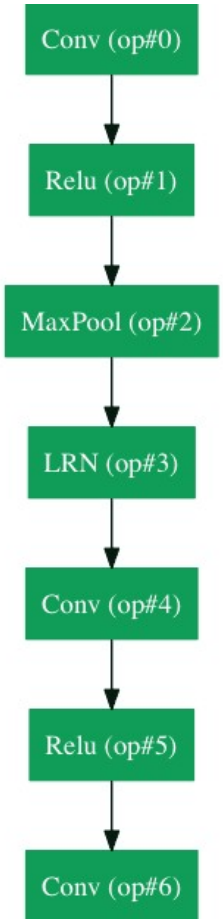
ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

Revolution of Depth



ImageNet Classification top-5 error (%)

Google LeNet (inception 2014)

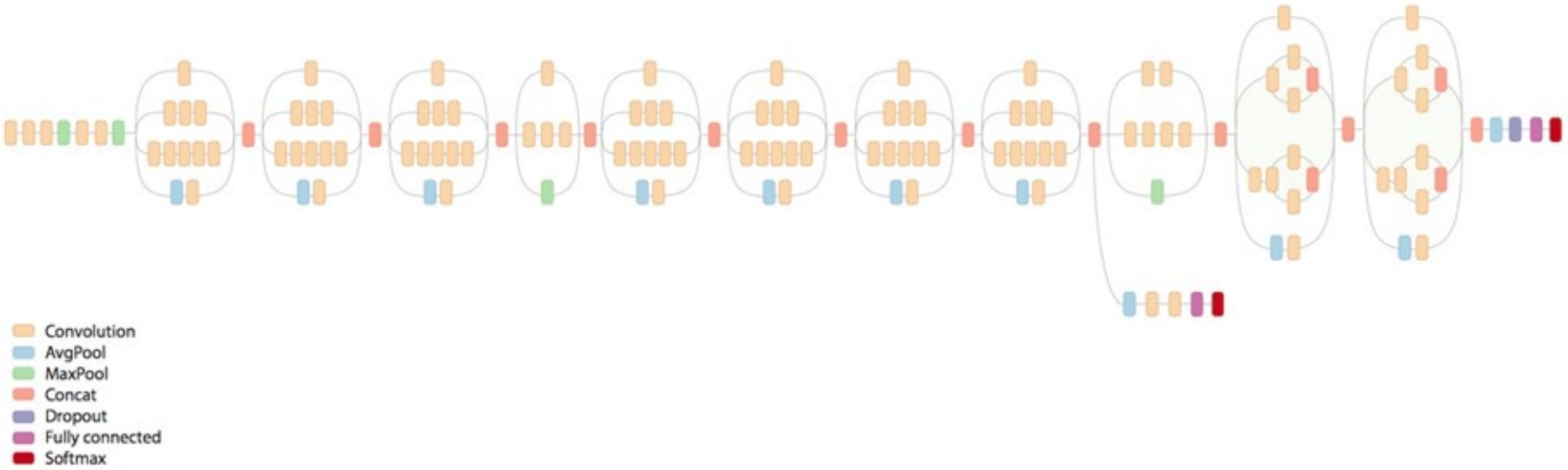


22 layers

6.67% error
ImageNet top 5

https://arxiv.org/abs/1409.4842

Google LeNet (inception 2014) – another view

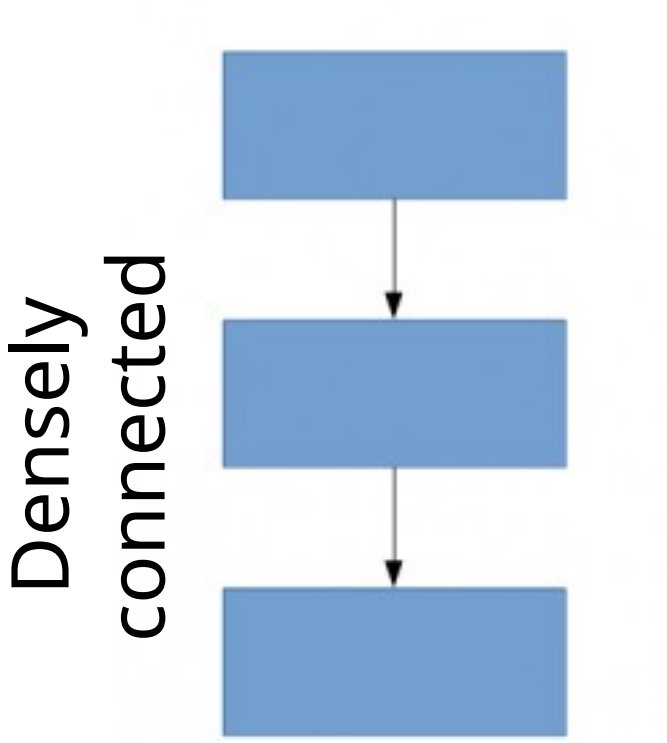


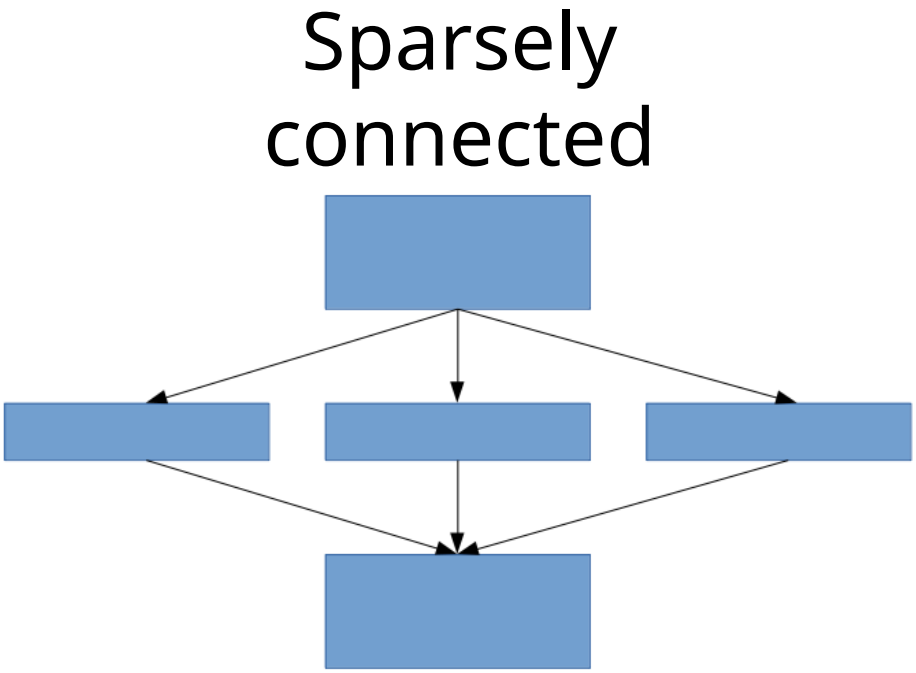
Another view of GoogLeNet's architecture.

Inception model idea:

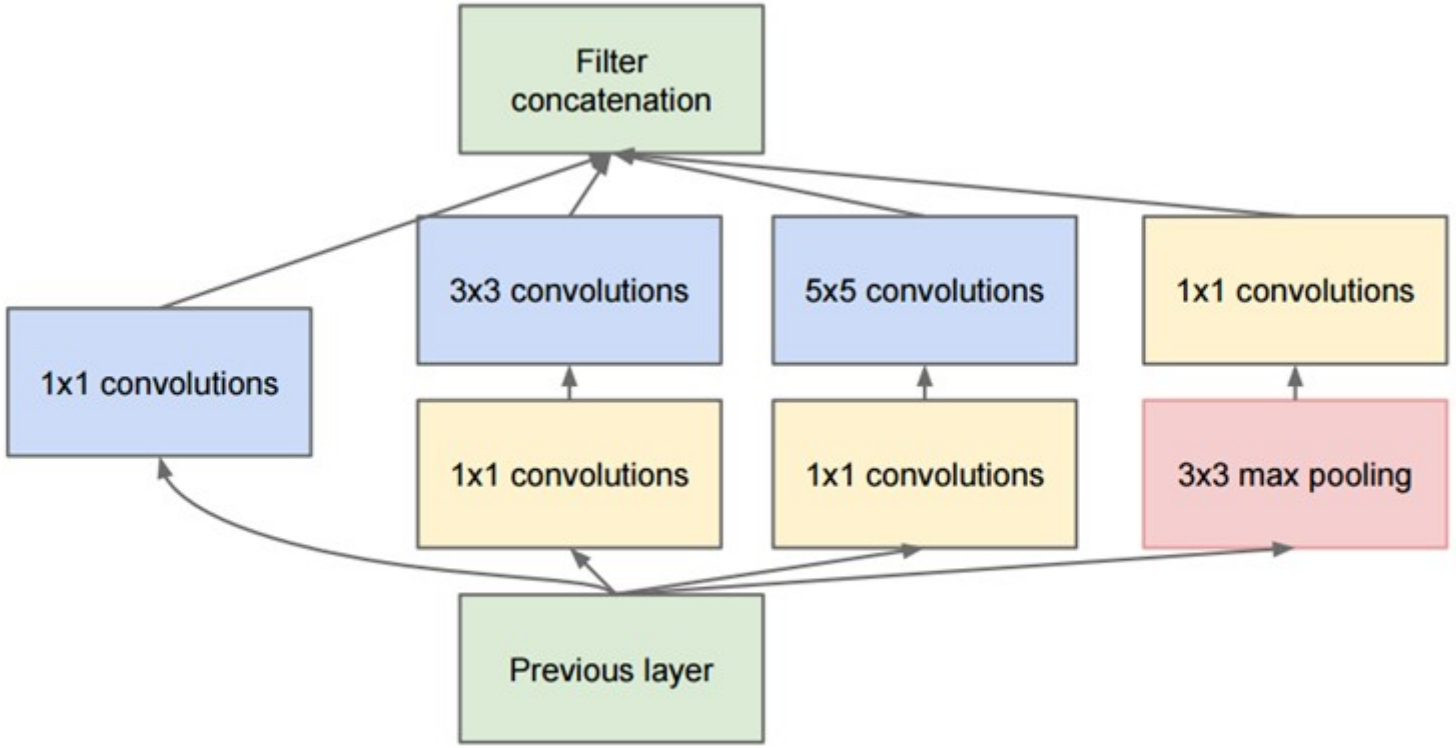
Go deeper while maintaining computational cost:

Achieved by replacing fully connected architectures with sparsely connected ones



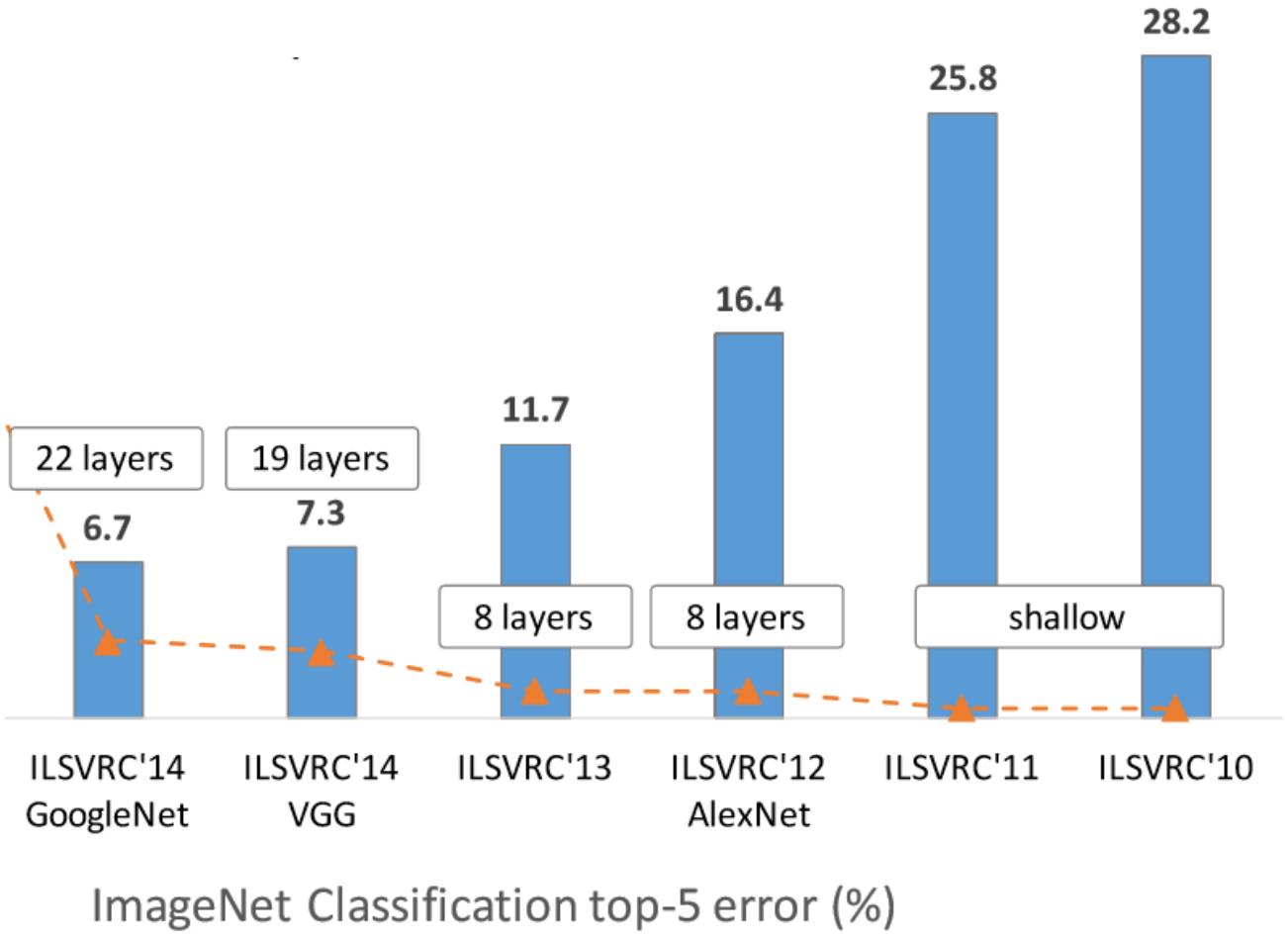


Sparsely connected layers: Parallel layers and 1x1 convolutions



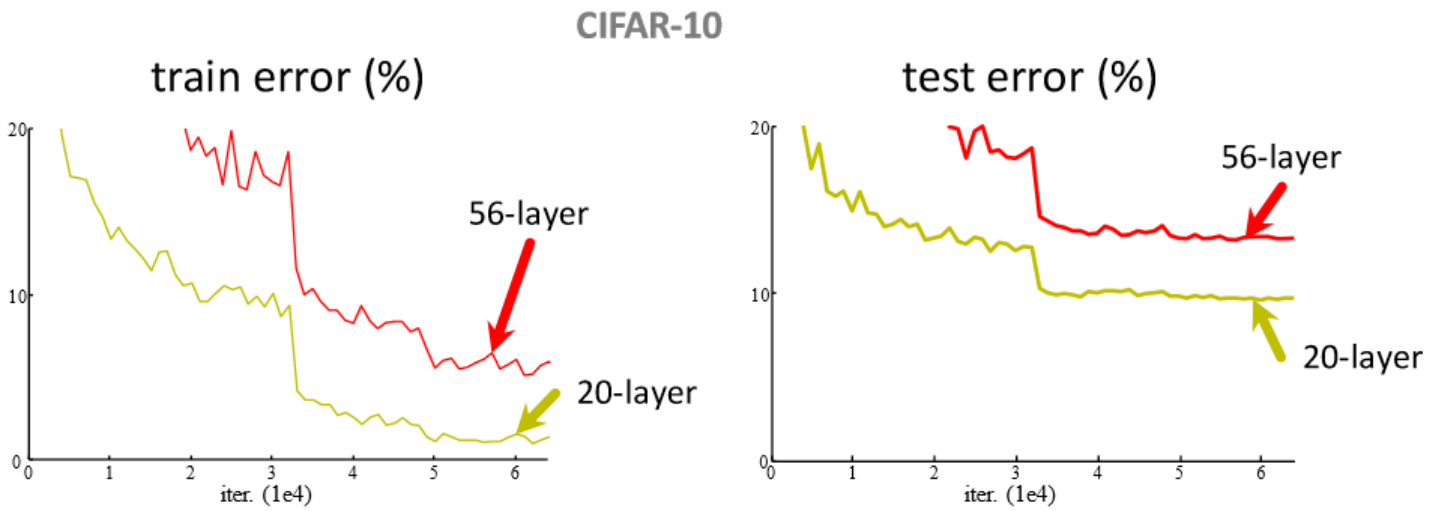
Full Inception module

Revolution of Depth



Problems with going deeper

Simply stacking layers?

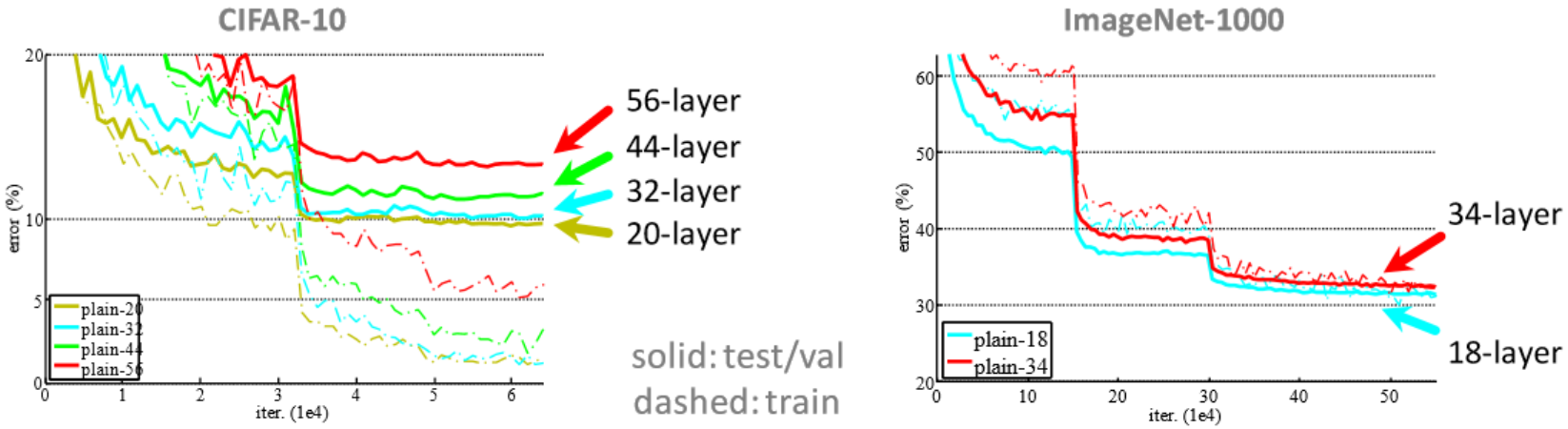


- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Problems with going deeper

Simply stacking layers?



- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Why simply stacking layers does not work:

Vanishing/exploding gradient problem

Backpropagation:

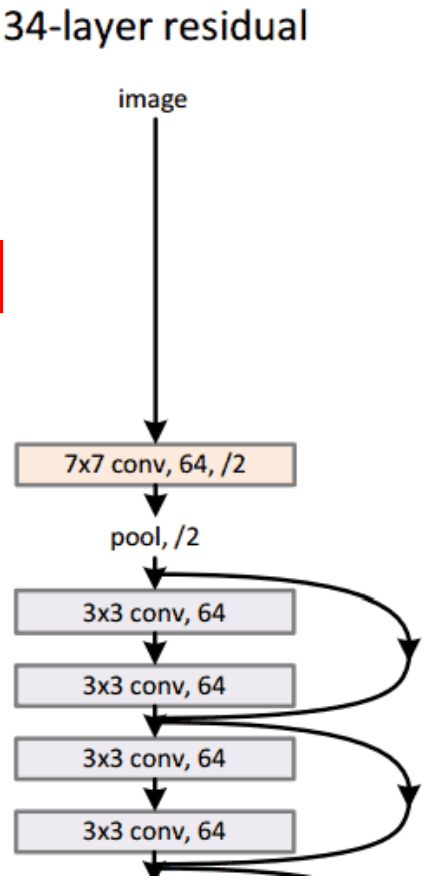
- Compute gradient update for every neuron which was involved in the output across layers

Involves chaining partial derivates over many layers!

- If derivative < 1, gradient gets smaller and smaller as we go deeper and deeper -> *vanishing gradients!*
- If derivative > 1, gradient gets larger and larger as we go deeper and deeper -> *exploding gradients!*

ResNet (He et al., 2015)

152-layer model



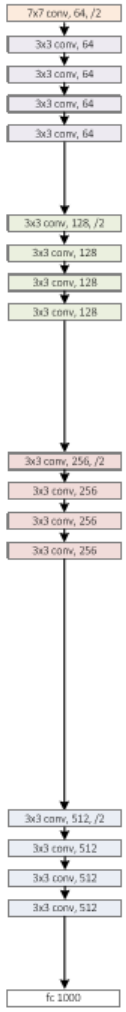
ResNet won ILSVRC 2015 with a top-5 error rate of 3.6%

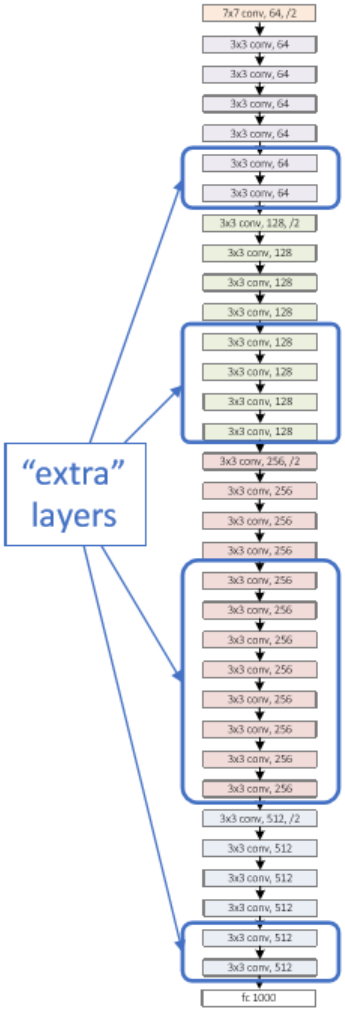
Depending on their skill and expertise, humans generally hover around a 5-10% error.

~~Superhuman performance!~~
But the task is arguably not well defined.

http://kaiminghe.com/cvpr16resnet/cvpr2016_deep_residual_learning_kaiminghe.pdf

a shallower model (18 layers)





a deeper counterpart (34 layers)

- Richer solution space
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

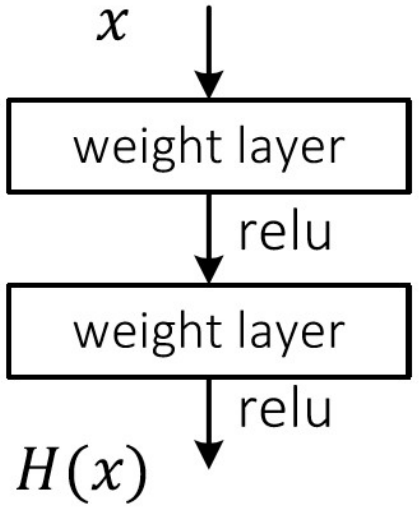
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

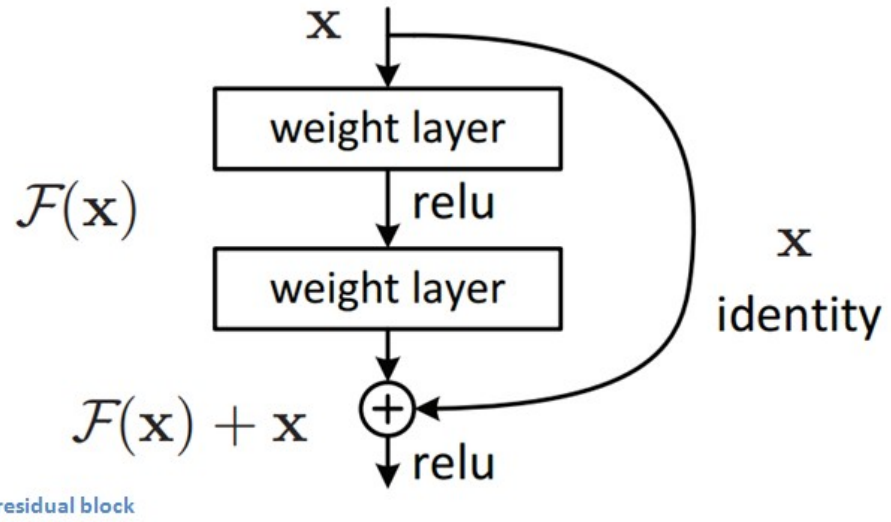
Vanishing/exploding gradient solution

Regular Net

Residual Unit

any two stacked layers

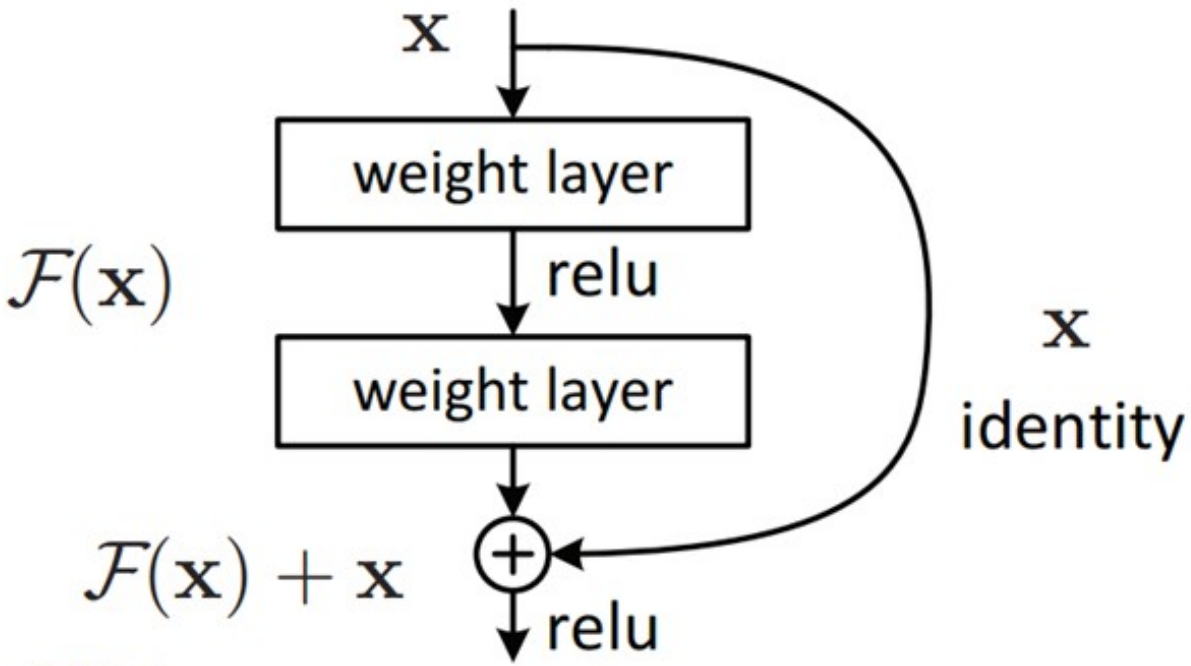




A residual block

Residual Unit

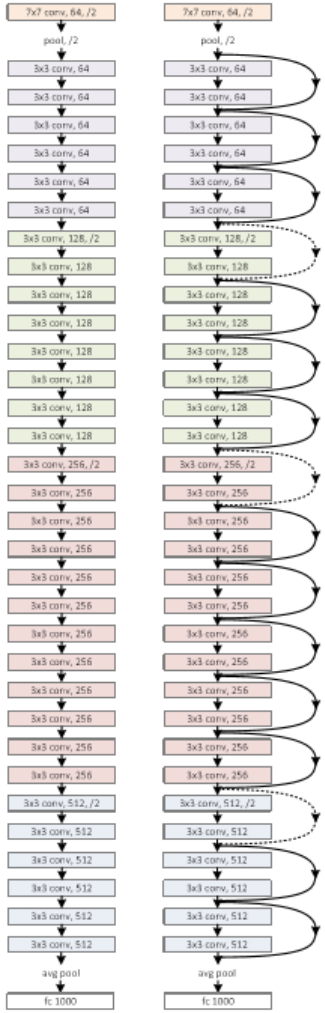
The inputs of a lower layer is made available to a node in a higher layer.



A residual block

Network “Design”

plain net

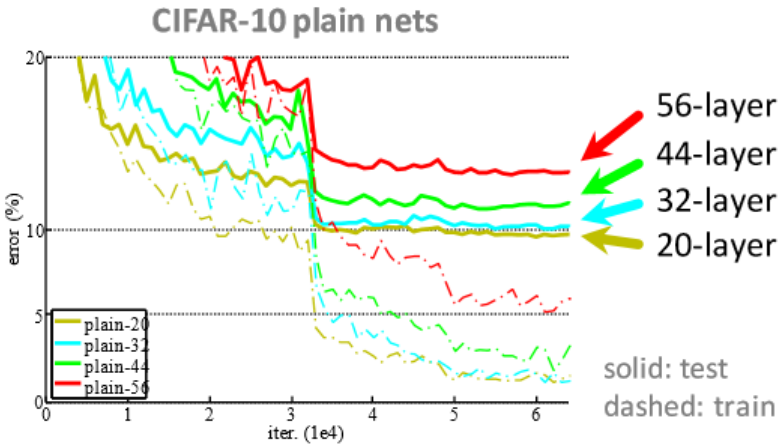


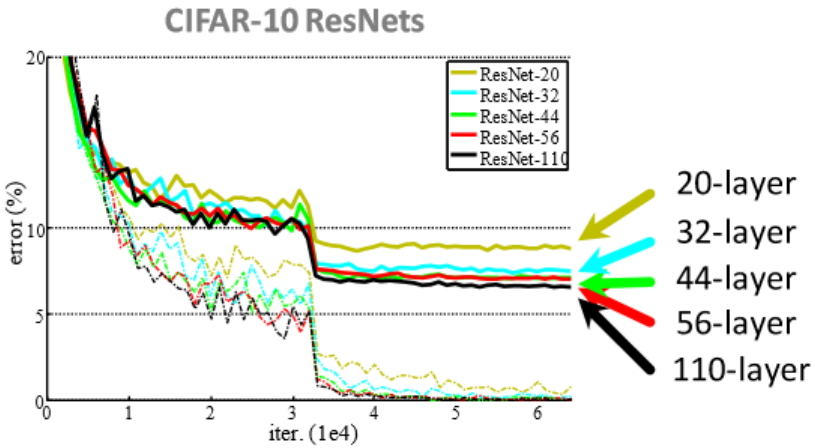
ResNet

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ResNet results

CIFAR-10 experiments

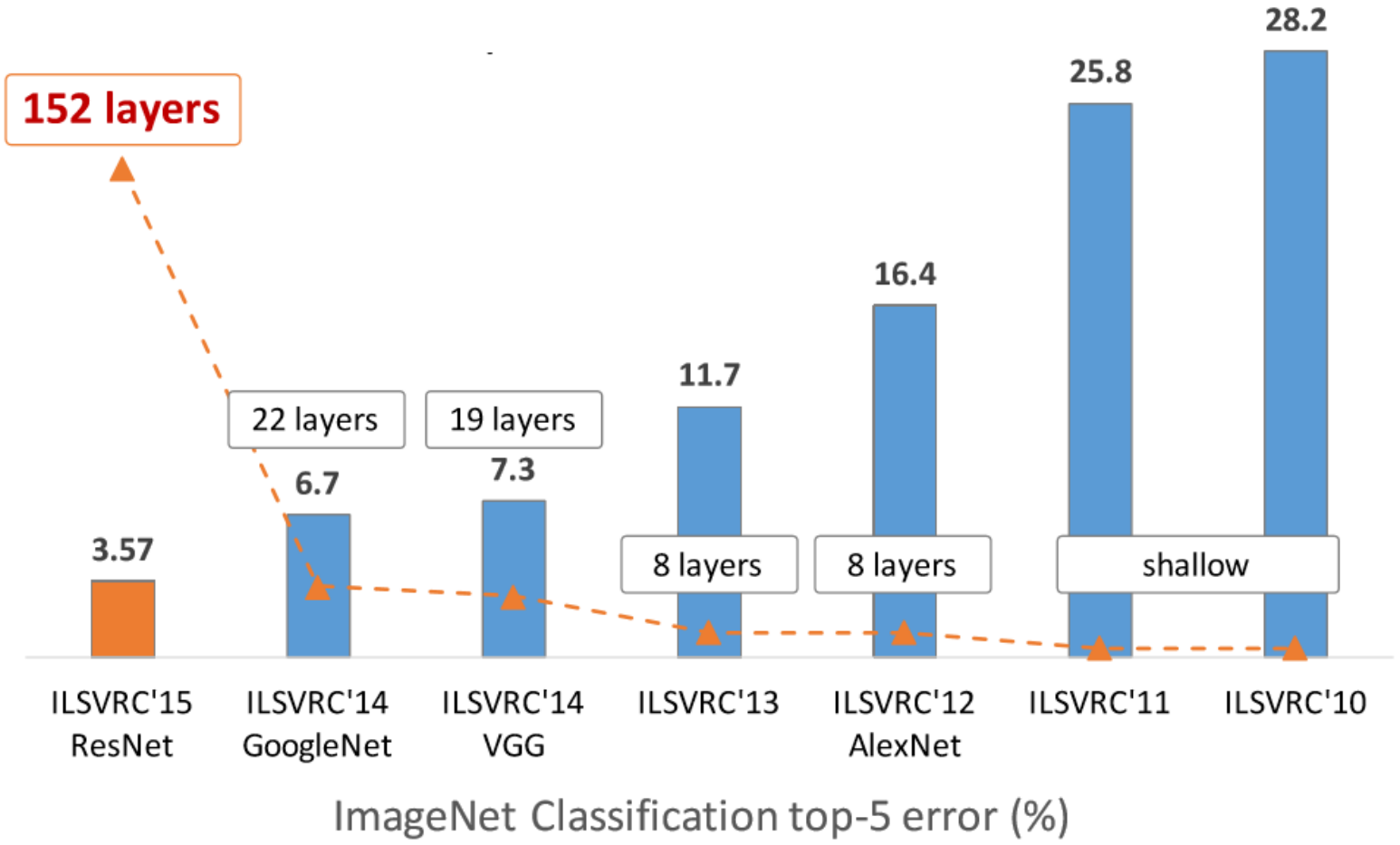




- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

R-CNN: Region-based CNN

R-CNN: Region-based CNN

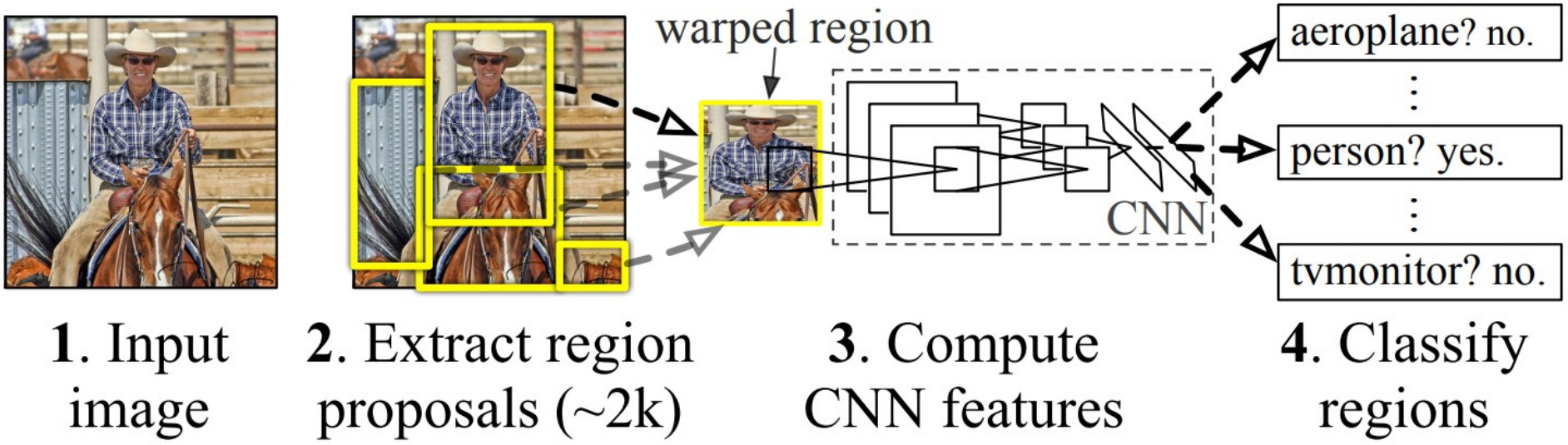


Figure: Girshick et al.

Reading material:

https://d2l.ai/chapter_computer-vision/rcnn.html

Efficient region proposals?

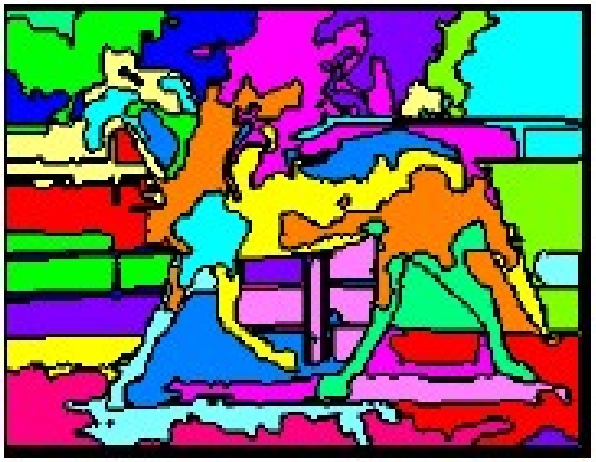
- Brute force on 1000x1000 = 250 billion rectangles
  - Testing the CNN over each one is too expensive
- Let’s use B.C. vision! Before CNNs
  - Hierarchical clustering for segmentation

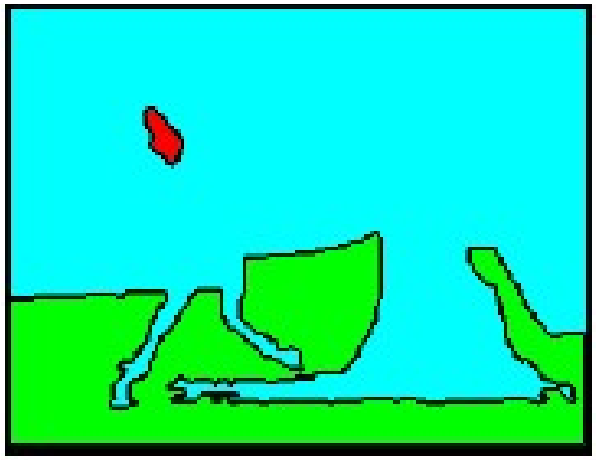
Use segmentation results for region proposals (instead of exhaustive search).

Uijlings et al., 2012, Selection Search

Thanks to Song Cao

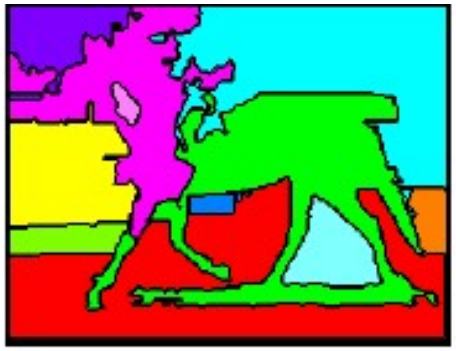
Remember clustering for segmentation

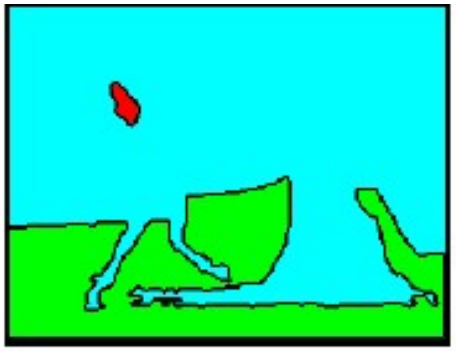




Oversegmentation

Undersegmentation





Hierarchical Segmentations

http://www.cs.cornell.edu/courses/cs7670/2014sp/slides/VisionSeminar14.pdf

Going through the details of selective search:

Original paper:

http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

presentations about the paper:

http://vision.stanford.edu/teaching/cs231b_spring1415/slides/ssearch_schuyler.pdf
http://www.cs.cornell.edu/courses/cs7670/2014sp/slides/VisionSeminar14.pdf

Cluster low-level features

- Define similarity on color, texture, size, ‘fill’
- Greedily group regions together by selecting the pair with highest similarity
  - Until the whole image become a single region
- Draw a bounding box around each one
  - Into a hierarchy

R-CNN: Region-based CNN

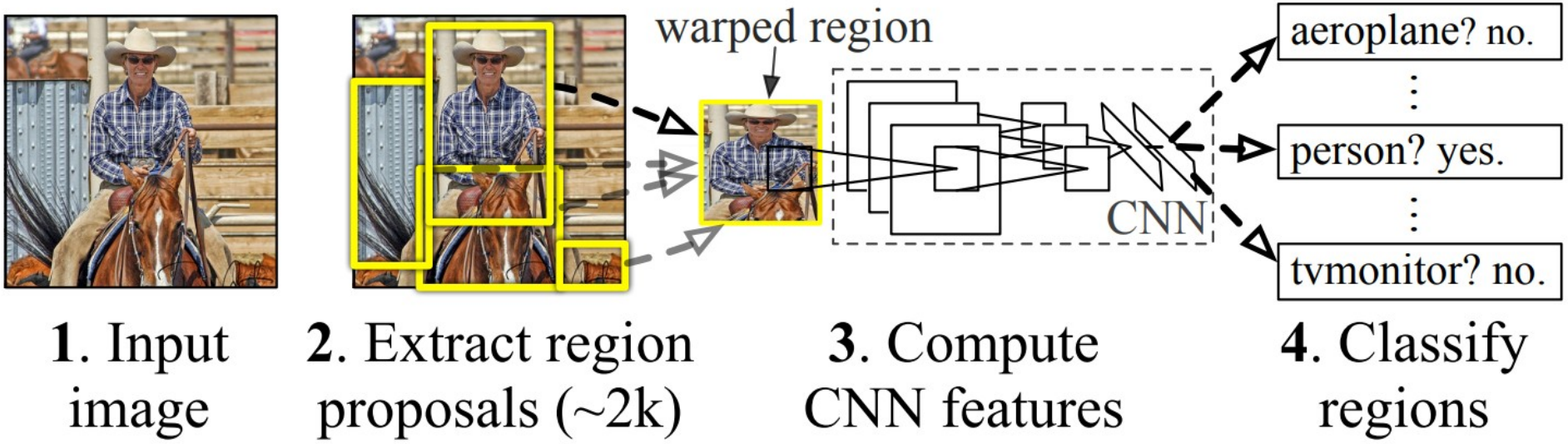
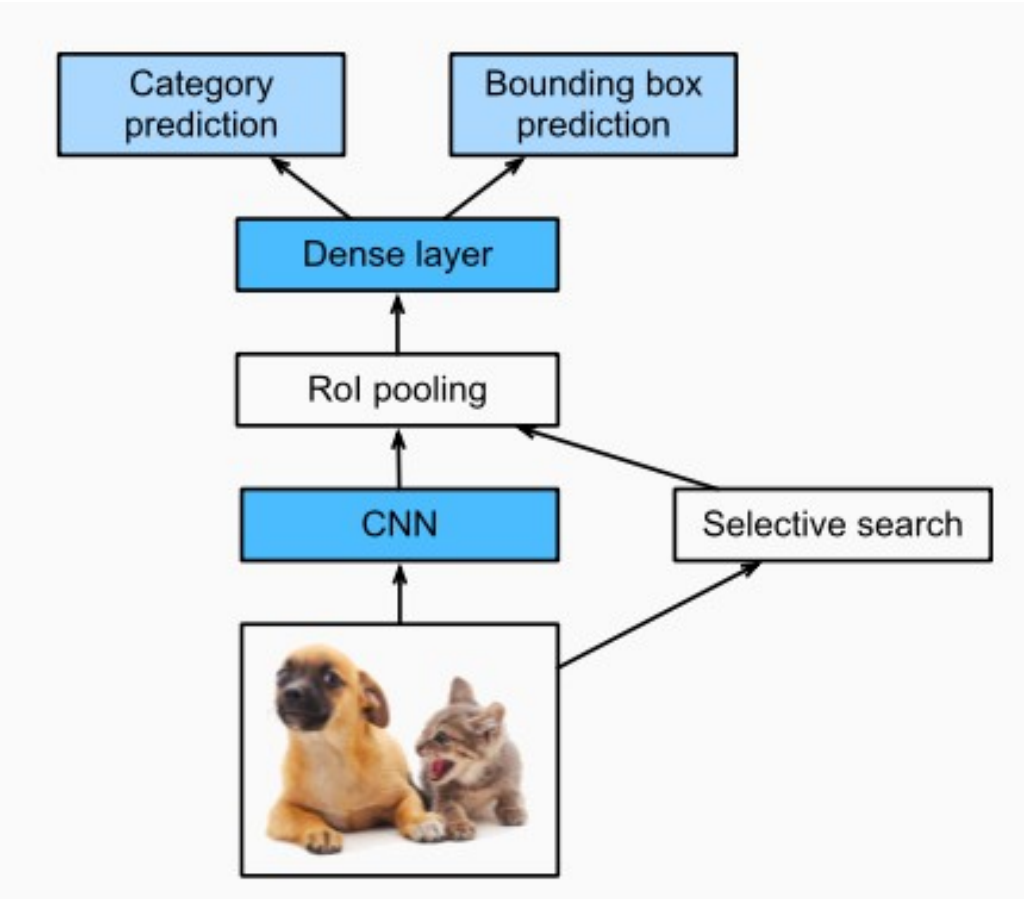


Figure: Girshick et al.

10,000 proposals with recall 0.991 is better than ~billions but still takes 17 seconds per image to generate them.

Then we have to perform classification on each one!

Fast R-CNN

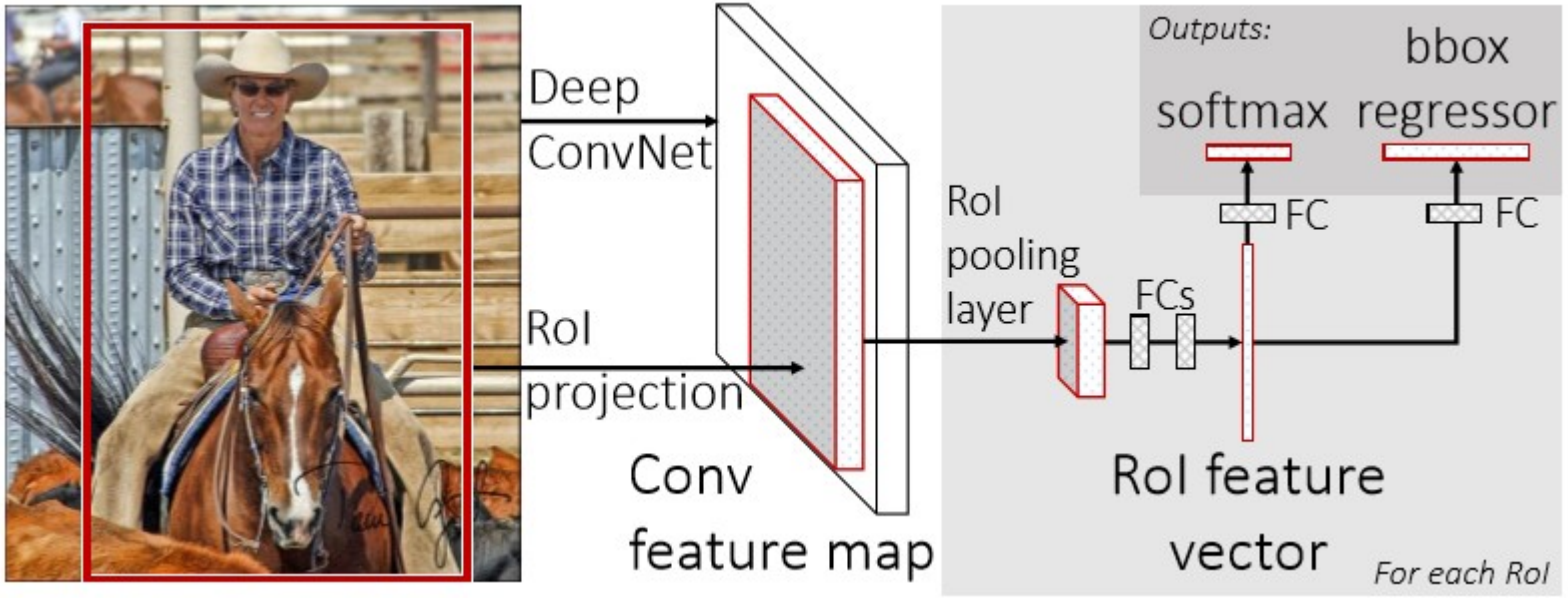


Note: Calling your thing ‘fast’ in computing is not descriptive. In three years, it will no longer be fast, and you will look silly.

https://d2l.ai/chapter_computer-vision/rcnn.html

Fast R-CNN

Figure: Girshick et al.



- Convolve whole image into feature map (many layers; abstracted)
- For each candidate RoI:
  - Squash feature map weights into fixed-size ‘RoI pool’
  - Divide RoI into H x W subwindows, e.g., 7 x 7, and max pool
  - Learn classification on RoI pool with own fully connected layers (FCs)
  - Output classification (softmax) + bounds (regressor)

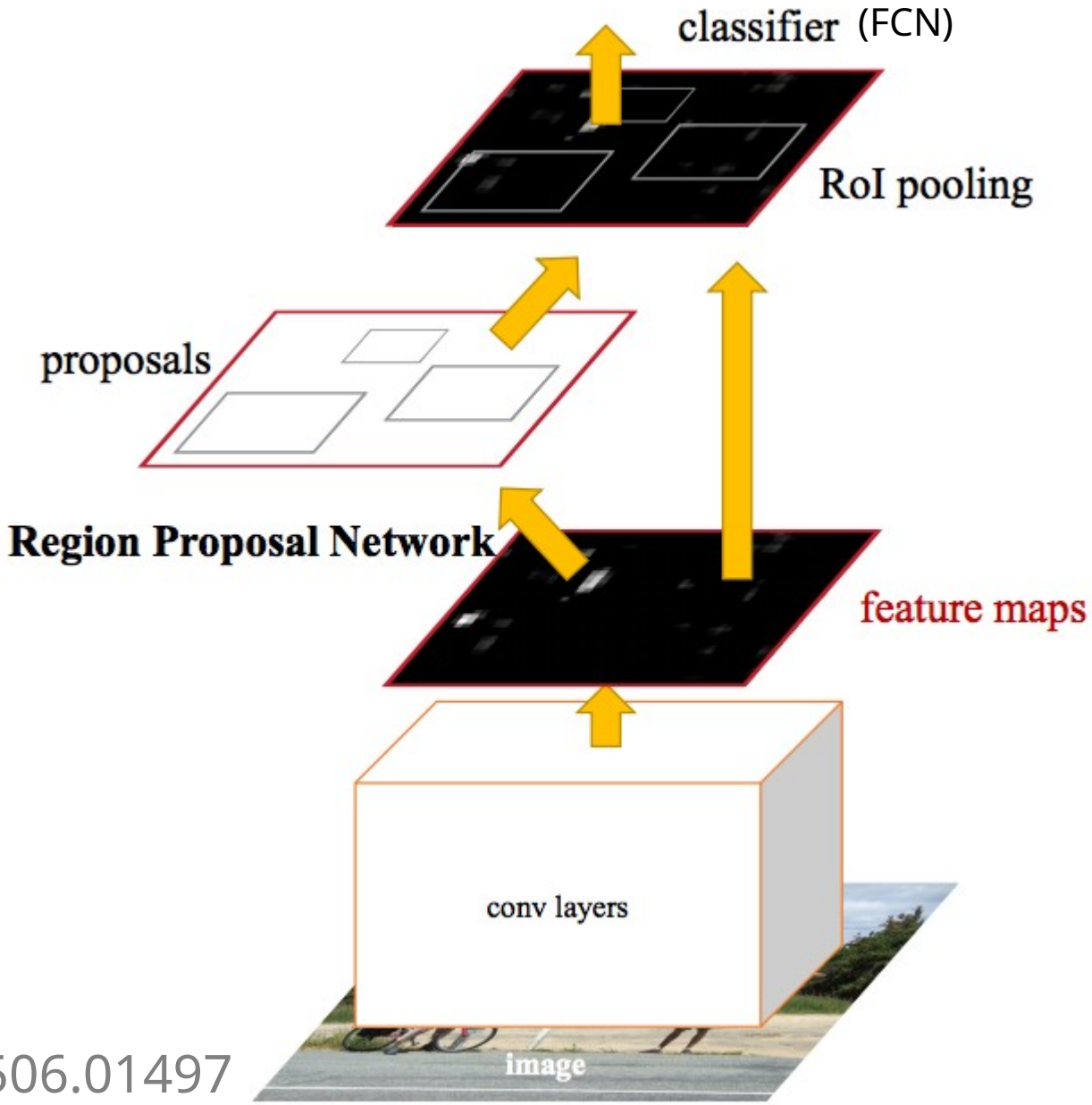
The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets.

Even more: Faster R-CNN

‘Region Proposal Network’ uses CNN feature maps.

Then, FCN on top to classify.

End to end object detection.



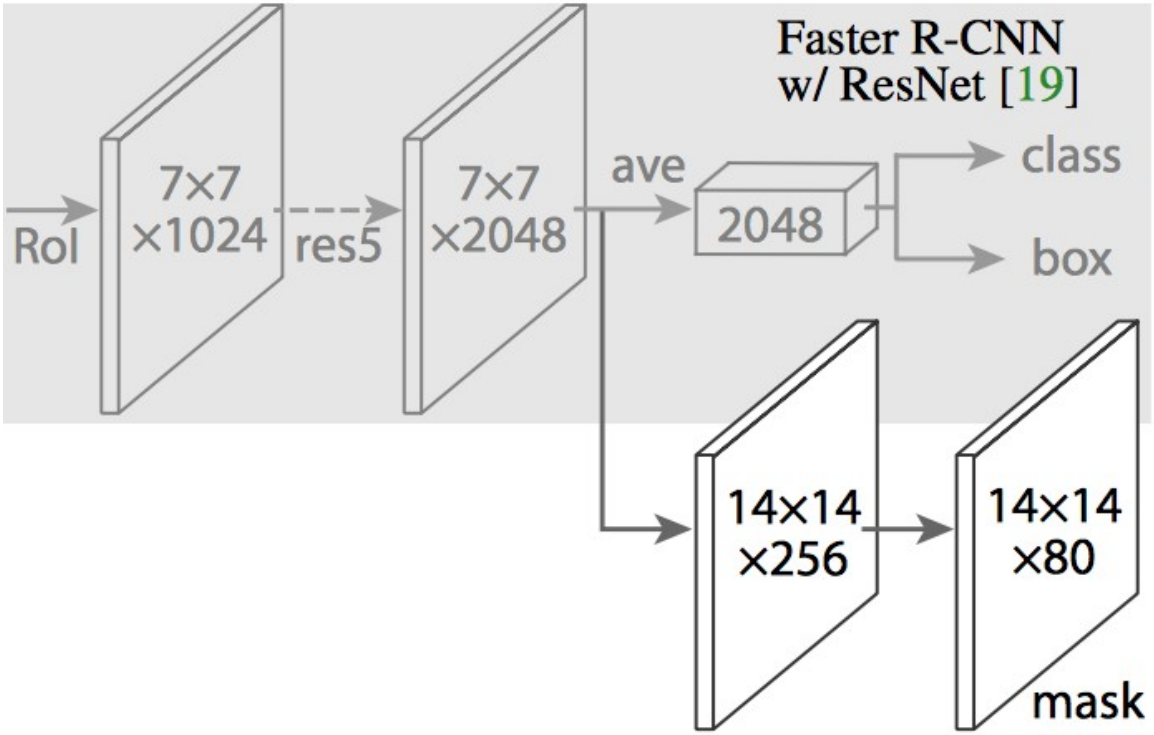
Ren et al. 2016
https://arxiv.org/abs/1506.01497

Even more! Mask R-CNN

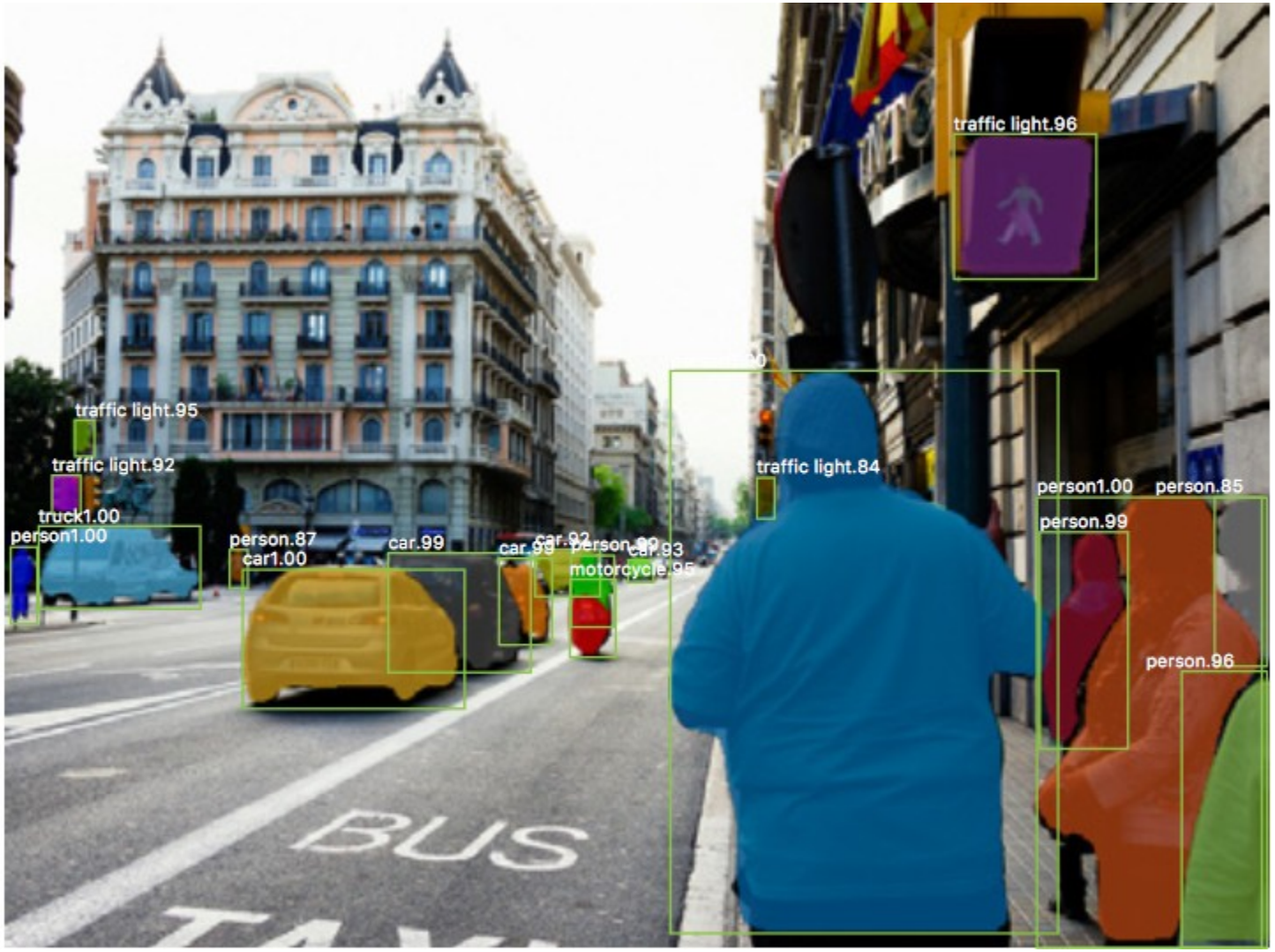
Extending Faster R-CNN for Pixel Level Segmentation

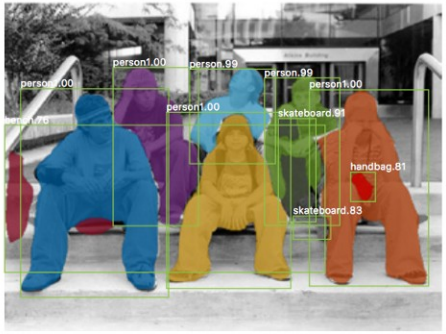
He et al. - https://arxiv.org/abs/1703.06870

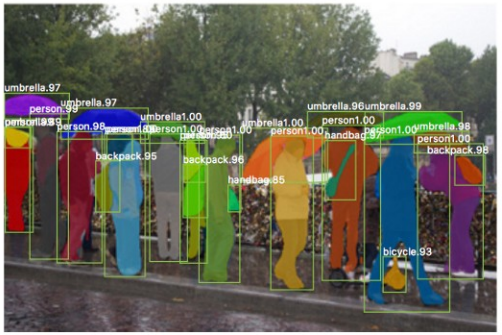
Add new training data: segmentation masks

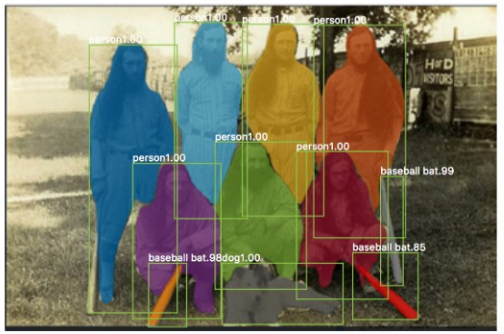


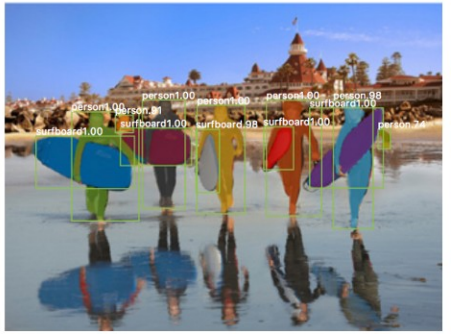
Second output which is segmentation mask


traffic light.96
traffic light.95
traffic light.92
truck1.00
person1.00
person.87
car1.00
car.99
car.99
car.92
person.99
car.93
motorcycle.95
traffic light.84
person1.00
person.85
person.99
person.96
BUS


person1.00
person.99
person1.00
person1.00
skateboard.81
handbag.81
skateboard.83

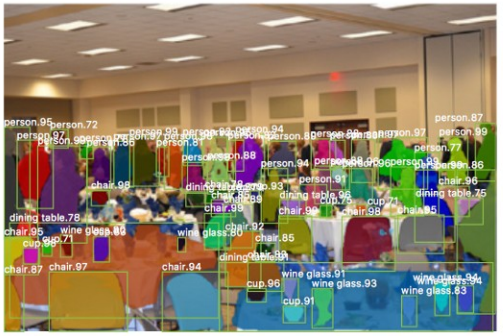

umbrella.97
umbrella.97
umbrella1.00
umbrella.96
umbrella.99
person1.00
backpack.95
backpack.96
handbag.85
backpack.98
bicycle.93

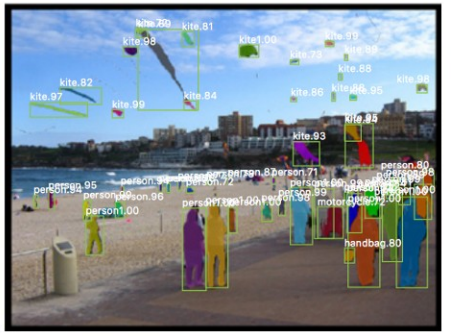

person1.00
person1.00
person1.00
person1.00
baseball bat.99
baseball bat.98
dog1.00
baseball bat.85


person1.00
surfboard1.00
surfboard1.00
surfboard1.00
surfboard1.00


zebra.99
zebra.99
zebra.88
zebra1.00
zebra1.00

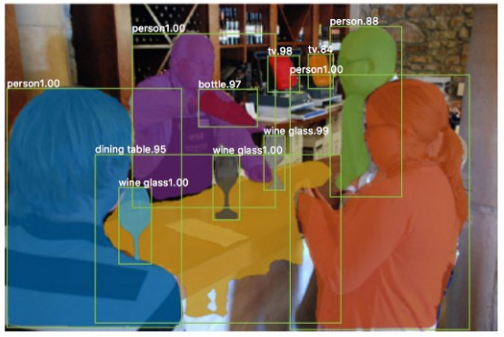

person.99
person.99
person1.00
person.82
person1.00
frisbee1.00
person.80


person.95
person.72
person.87
person.99
person.77
chair.95
chair.96
dining table.78
dining table.85
chair.87
chair.97
chair.94
wine glass.91
wine glass.93
cup.96
cup.91


kite.81
kite.98
kite1.00
kite.82
kite.97
kite.99
kite.84
kite.93
person1.00
handbag.80

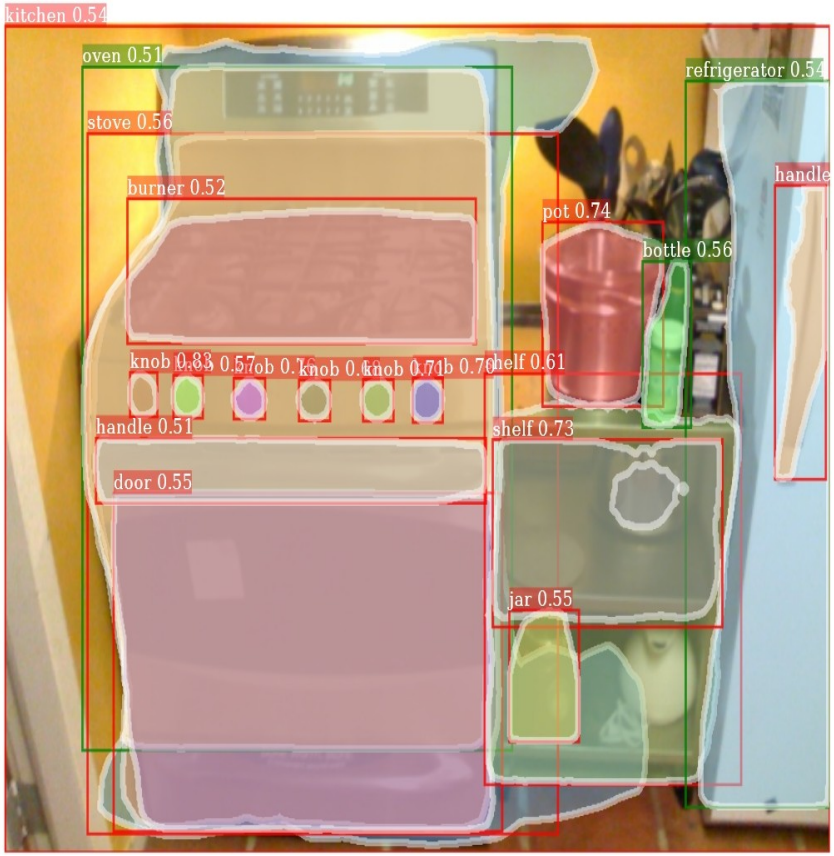

elephant1.00
elephant1.00
elephant1.00
elephant.97
elephant.99

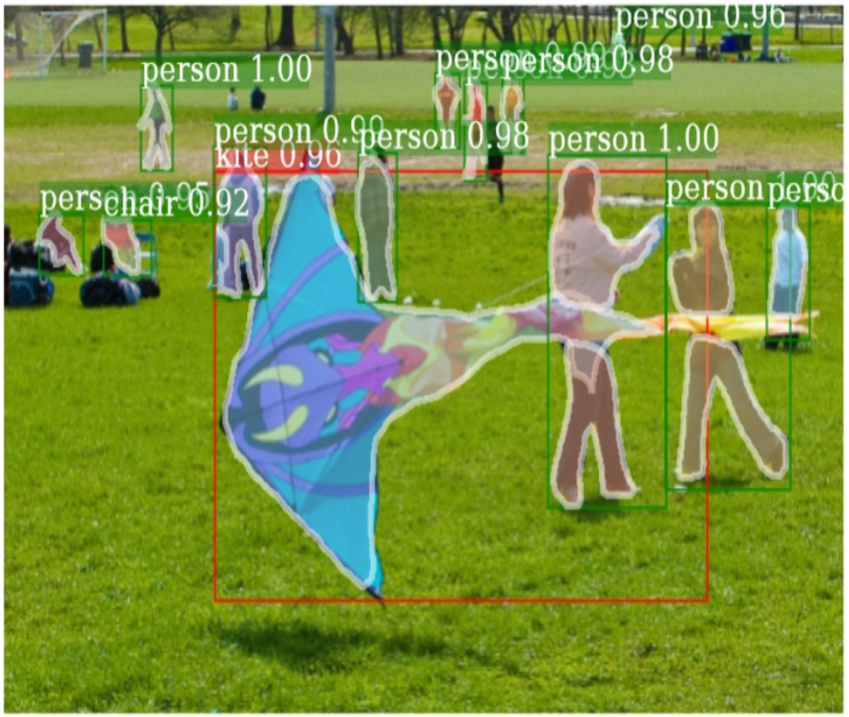

traffic light.99
traffic light.71
person.99
tie.85
handbag.88
handbag.99
handbag.88


person1.00
person.88
tv.98
person1.00
bottle.97
wine glass.99
dining table.95
wine glass1.00
wine glass1.00


potted plant.92
person1.00
person.88
bird.93
bench.97
handbag.73
cell phone.37

Learning to Segment Everything Mask X R-CNN





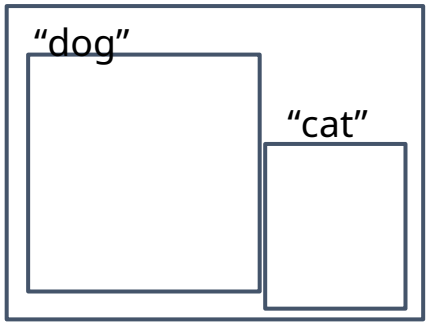
(Hu et al., 2018)

FCN: FULLY CONVOLUTIONAL NETWORKS

R-CNN does detection

many seconds

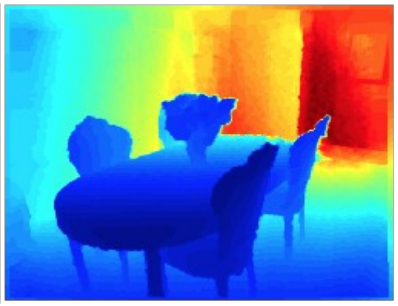
R-CNN

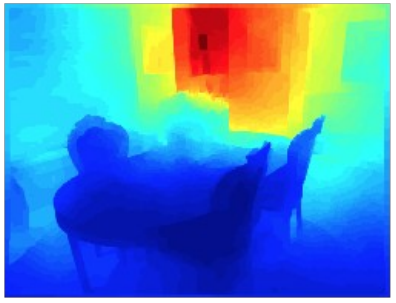


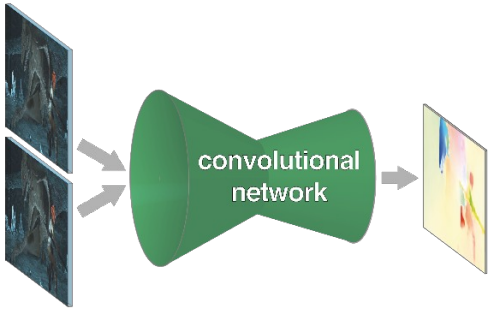
What if we want pixels out?

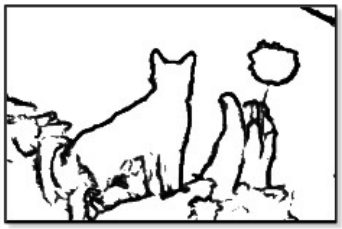
semantic segmentation

monocular depth estimation Eigen & Fergus 2015





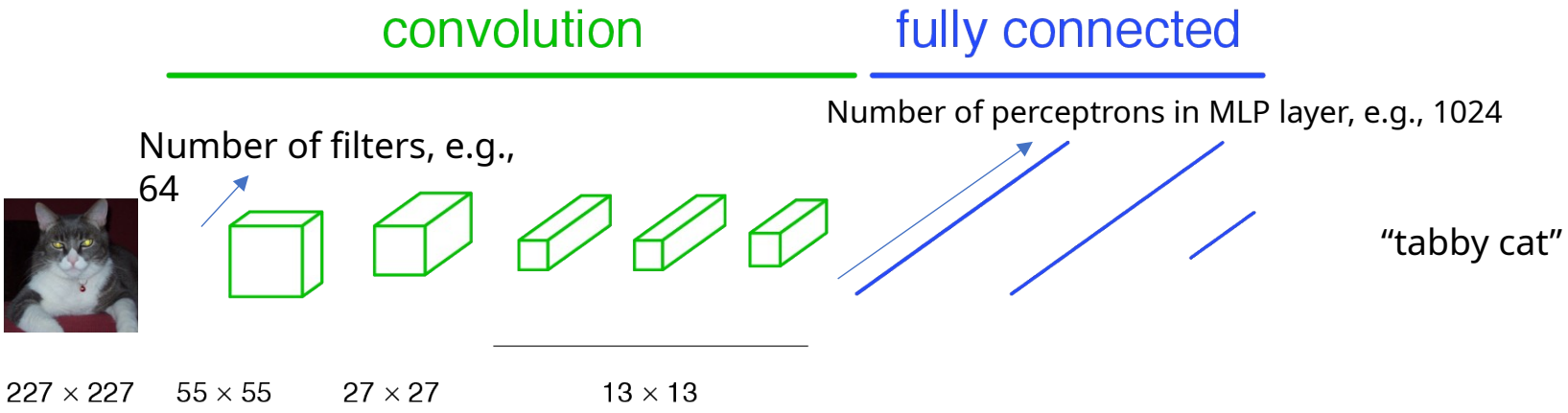


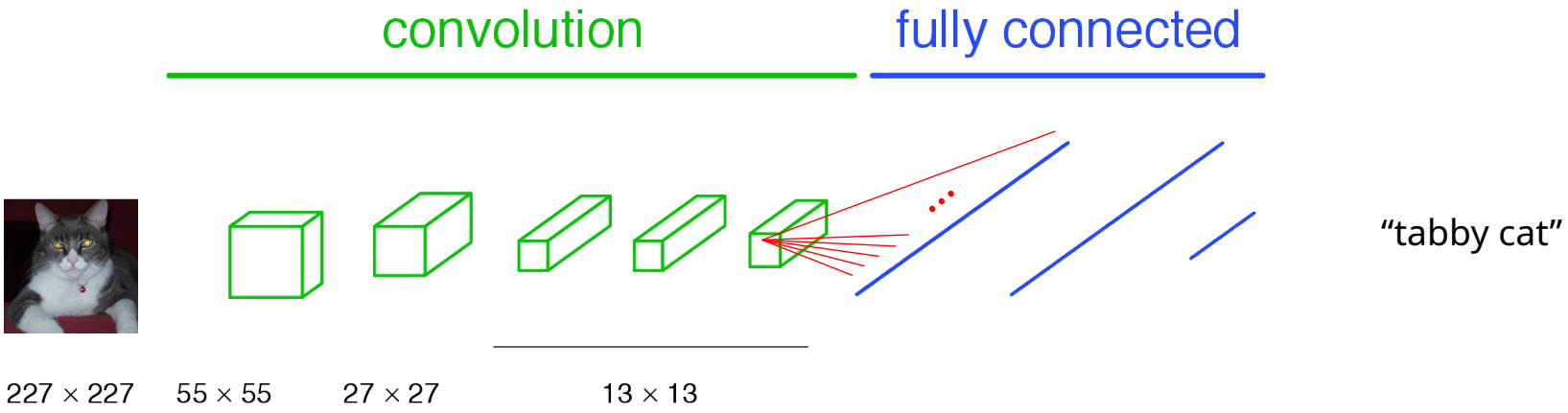


boundary prediction Xie & Tu 2015

[Long et al.]

A classification network…





The response of every kernel across all positions are attached densely to the array of perceptrons in the fully-connected layer.

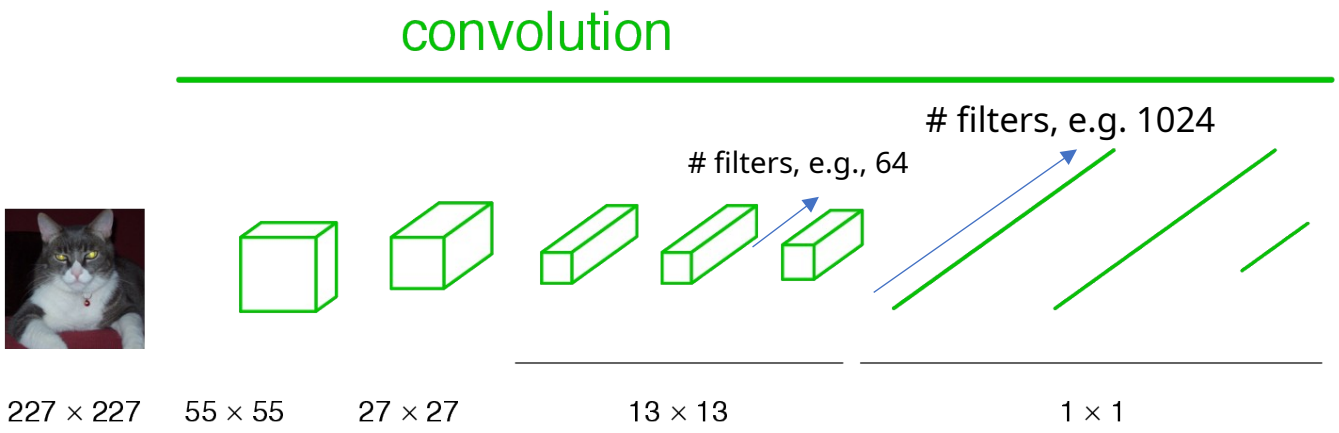
[Long et al.]

The response of every kernel across all positions are attached densely to the array of perceptrons in the fully-connected layer.

AlexNet parameters

Layer	Units	Weights
L_1 (Conv)	290,400	34,848
L_2 (Conv)	186,624	307,200
L_3 (Conv)	64,896	884,736
L_4 (Conv)	64,869	663,552
L_5 (Conv)	43,264	442,368
L_6 (Dense)	4096	37,748,736
L_7 (Dense)	4096	16,777,216
L_8 (Dense)	1000	4,096,000
Conv Subtotal	650,080	2,332,704
Dense Subtotal	9192	58,621,952
Total	659,272	60,954,656

Convolutionalization

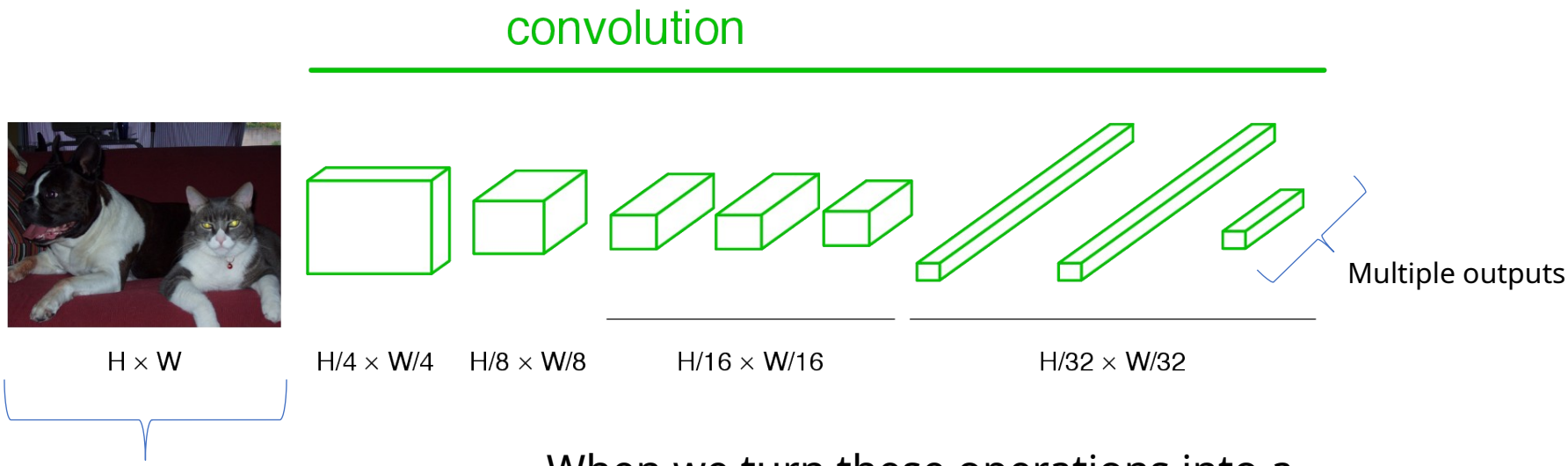


1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

e.g., 64x1x1 kernel, with shared weights over 13x13 output, x1024 filters = 11mil params.

[Long et al.]

Becoming fully convolutional



Arbitrary-sized image

When we turn these operations into a convolution, the 13x13 just becomes another parameter and our output size adjust dynamically.

Now we have a *vector/matrix* output, and our network acts itself like a complex filter.

[Long et al.]

Fully Convolutional Networks

Yann LeCun
6 April 2015 ·

Follow

In Convolutional Nets, there is no such thing as "fully-connected layers". There are only convolution layers with 1x1 convolution kernels and a full connection table.

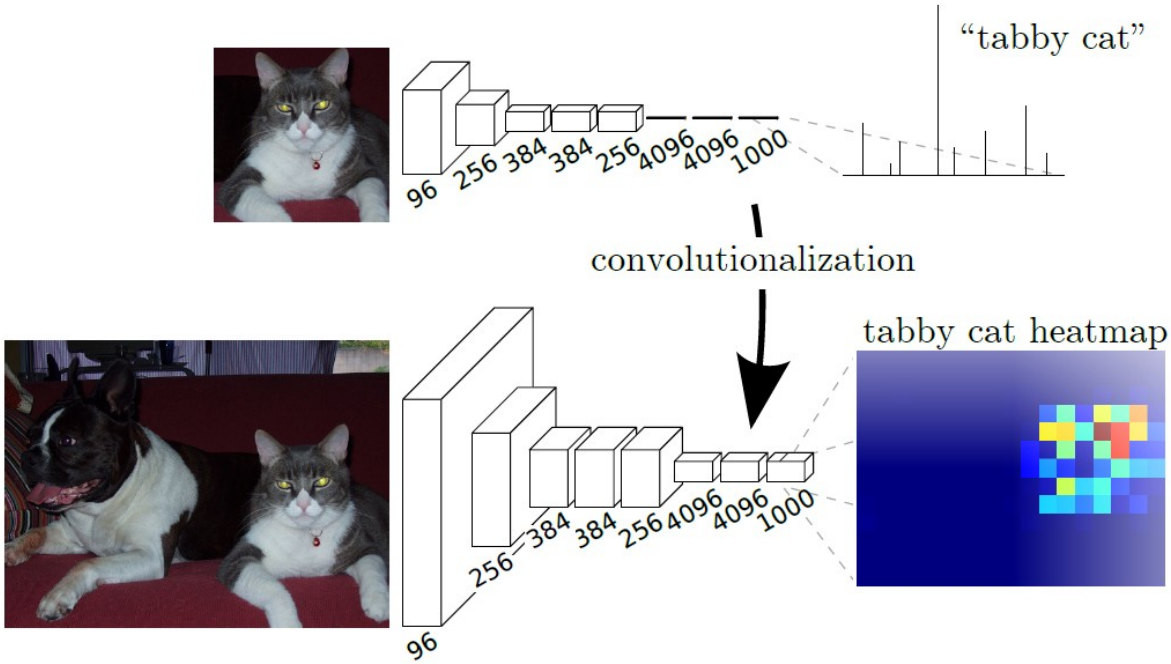
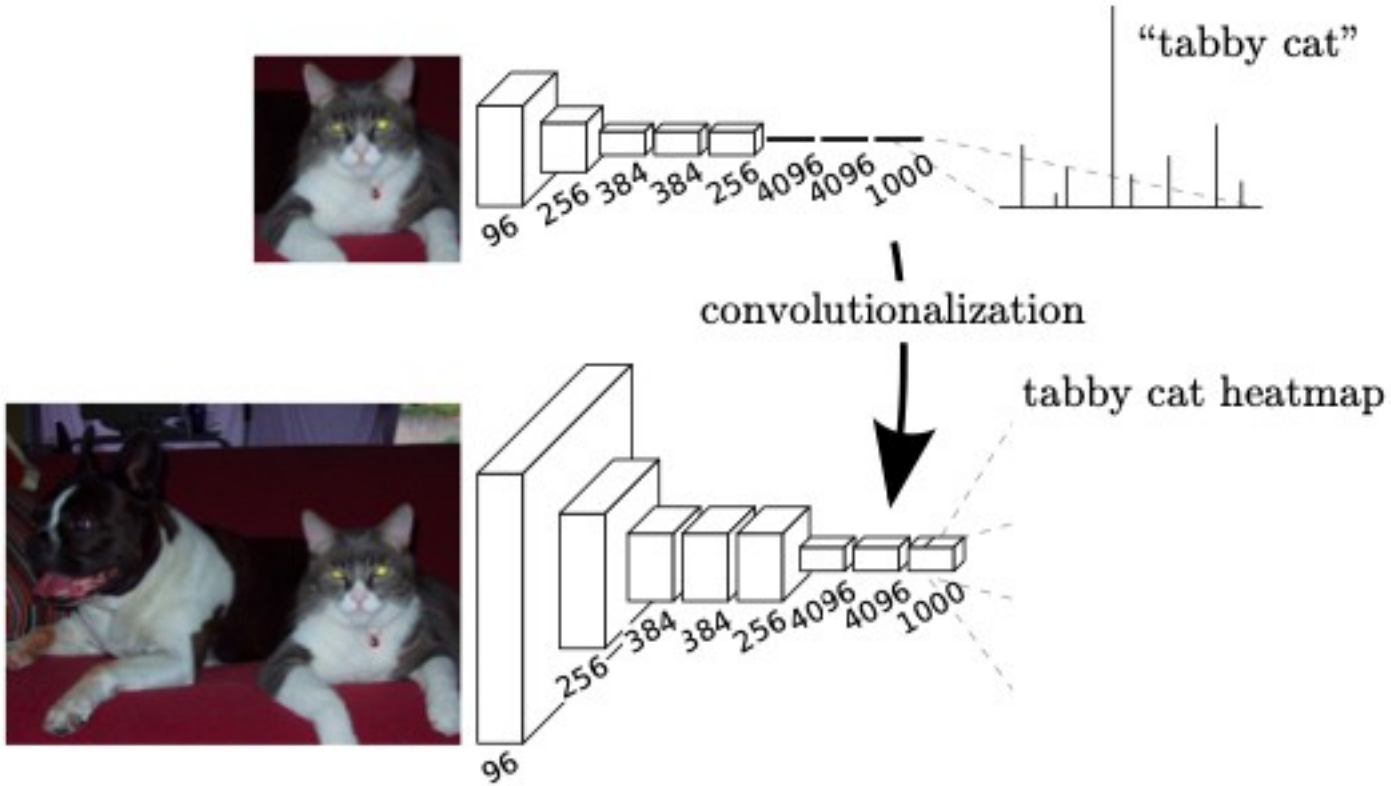


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Long, Shelhamer, and Darrell 2014

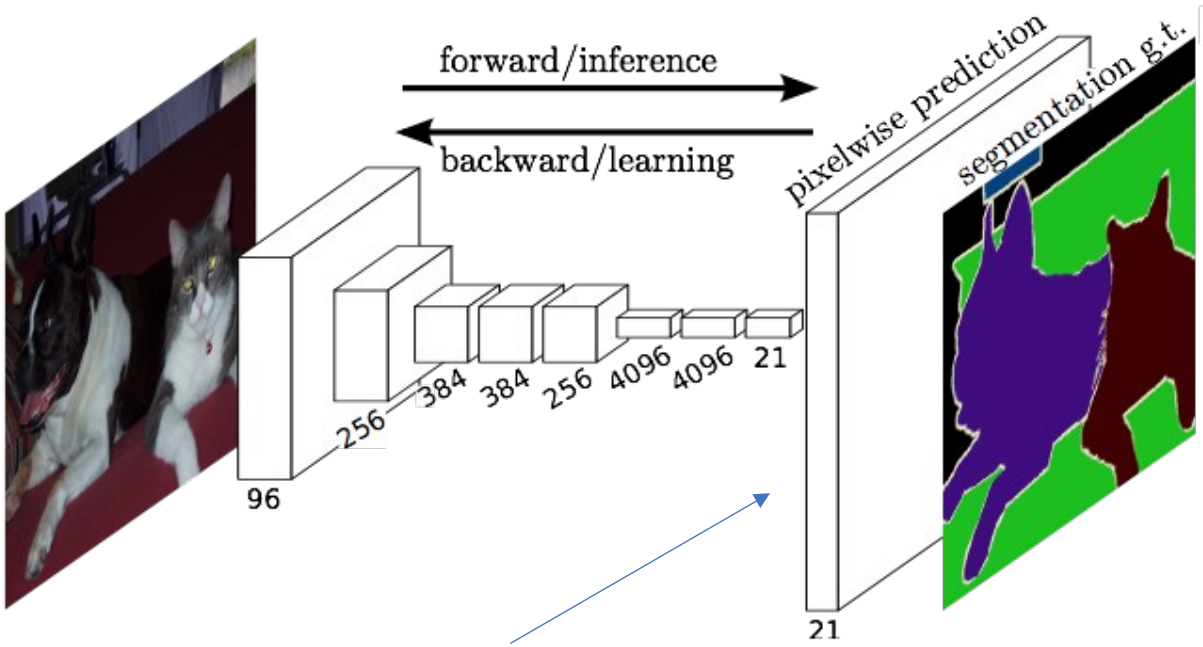
Fully Convolutional Networks



Differences from regular CNN

- Can naturally handle varying input image sizes
- Does not forego spatial information
- Output heat maps (the classification parameters is along the kernel's depth)

Fully Convolutional Networks for Semantic Segmentation



pixelwise output + loss

Some upsampling algorithm to return us to H x W

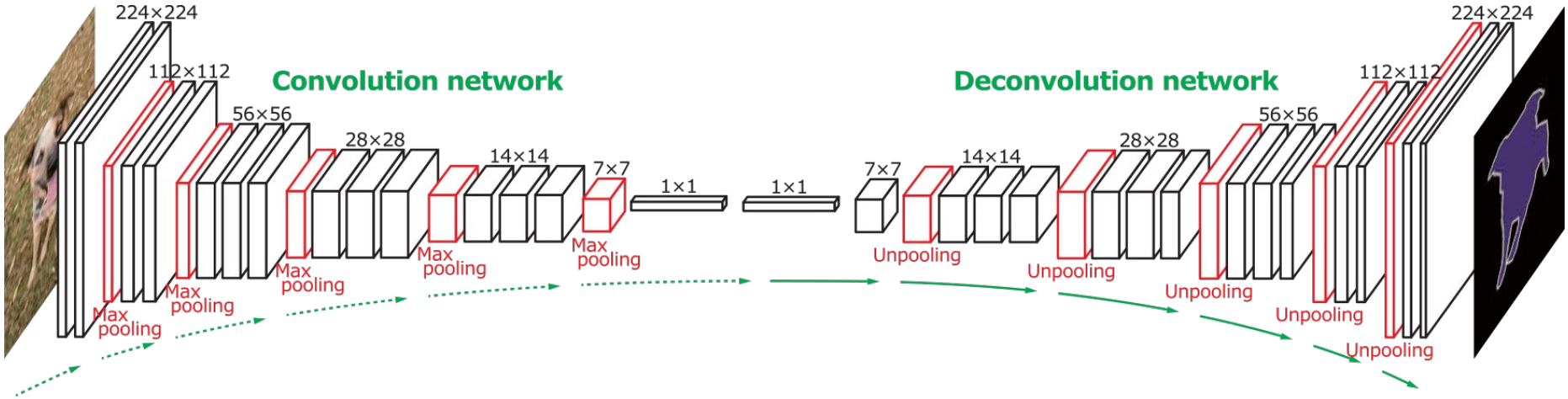
it's actually an upsampling *network*

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley

[CVPR 2015]

Slides from Long, Shelhamer, and Darrell

‘Deconvolution’ networks *learn to upsample*

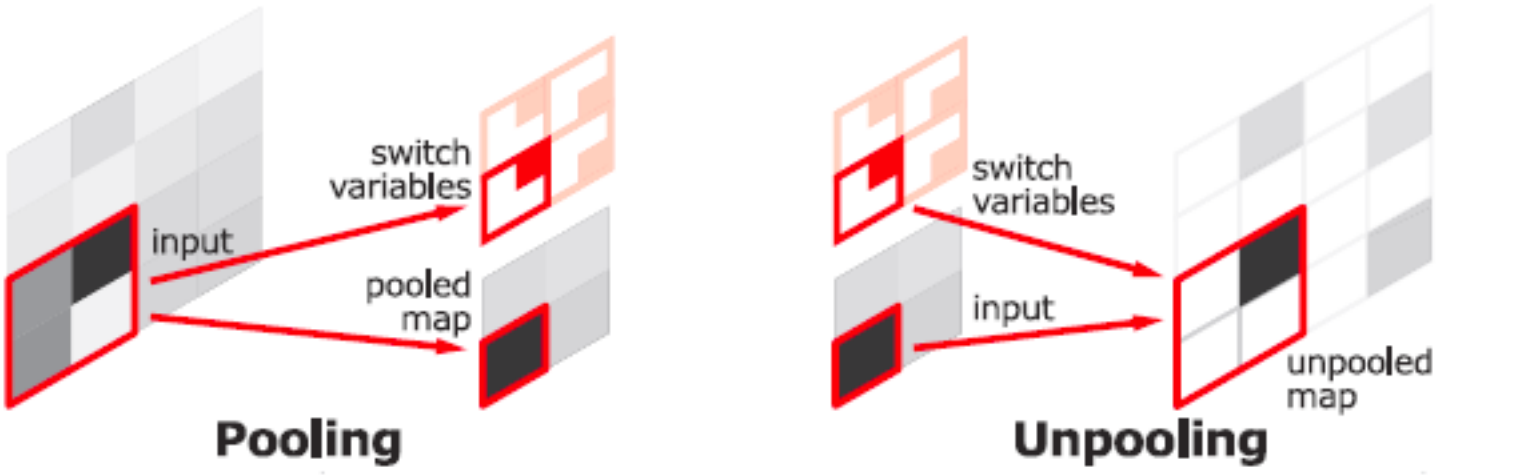


Often called “deconvolution”, but misnomer.
Not the deconvolution that we saw in deblurring -> that is division in the Fourier domain.

‘Transposed convolution’ is better.

Zeiler et al., Deconvolutional Networks, CVPR 2010
Noh et al., Learning Deconvolution Network for Semantic Segmentation, ICCV 2015

Unpooling layer

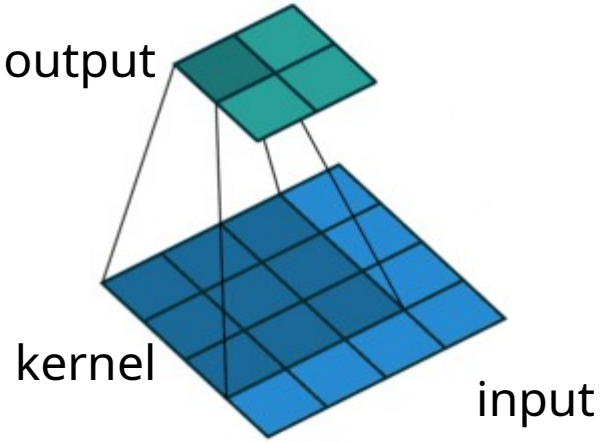


Remember positions when Pooling (Left), Reuse the position information during Unpooling (right)

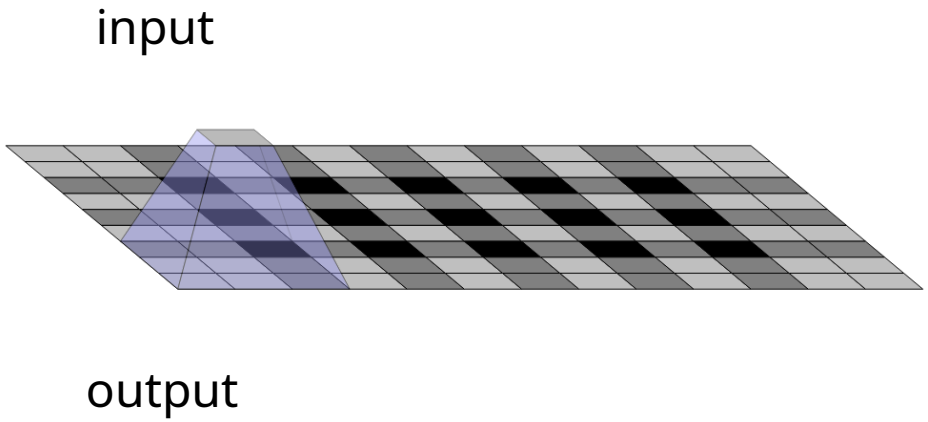
https://towardsdatascience.com
/review-deconvnet-unpooling-layer-semantic-segmentation-55cf8a6e380e

Upsampling with transposed convolution

Convolution



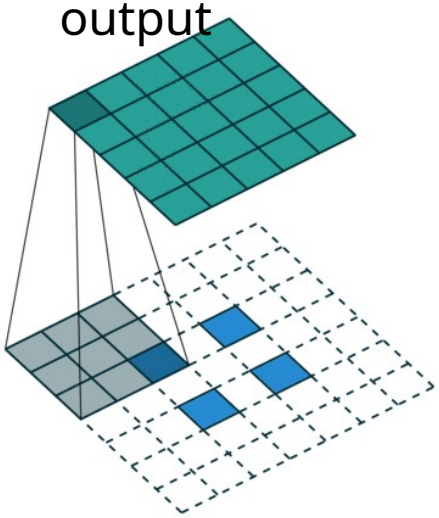
Transposed convolution smaller image
weighted sum of input x filter: ‘stamping’
kernel



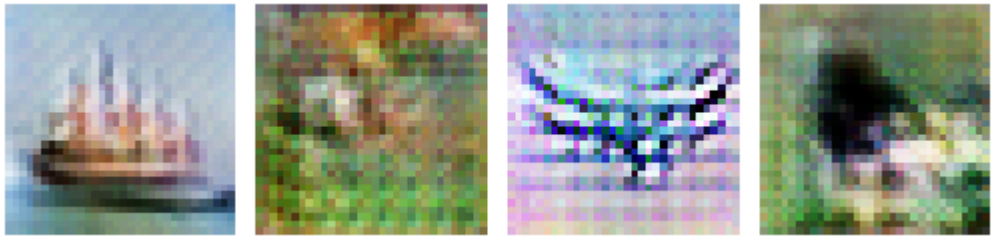
https://distill.pub/2016/deconv-checkerboard/

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

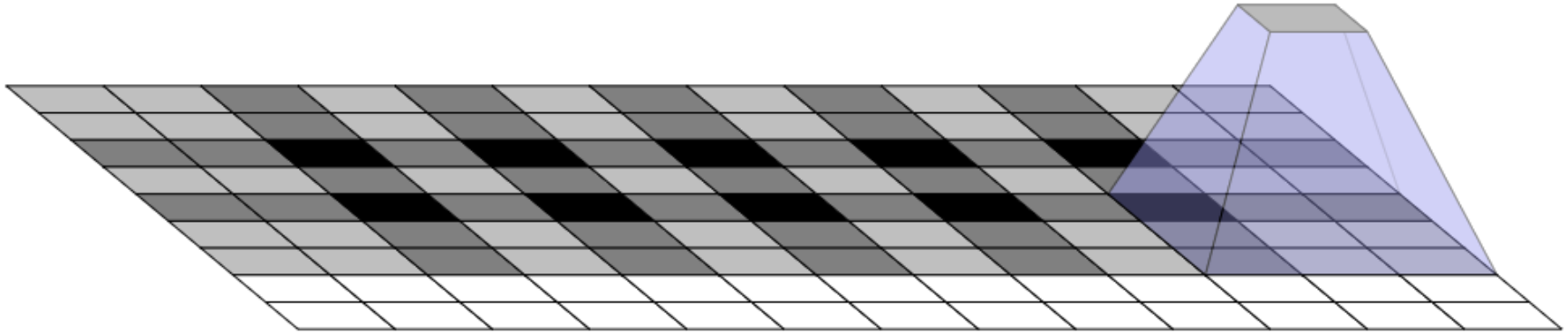
Is uneven overlap a problem?



Yes = causes grid artifacts

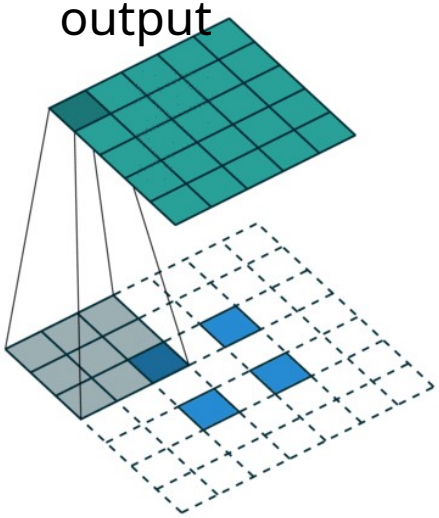


Could fix it by picking stride/kernel numbers which have no overlap...

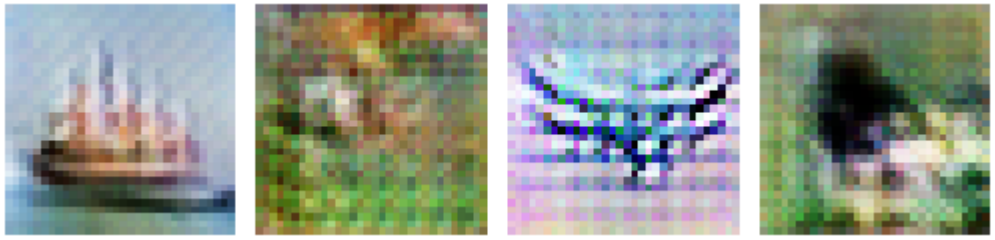


Is uneven overlap a problem

Uneven overlap across output



Yes = causes grid artifacts



Could fix it by picking stride/kernel numbers which have no overlap…

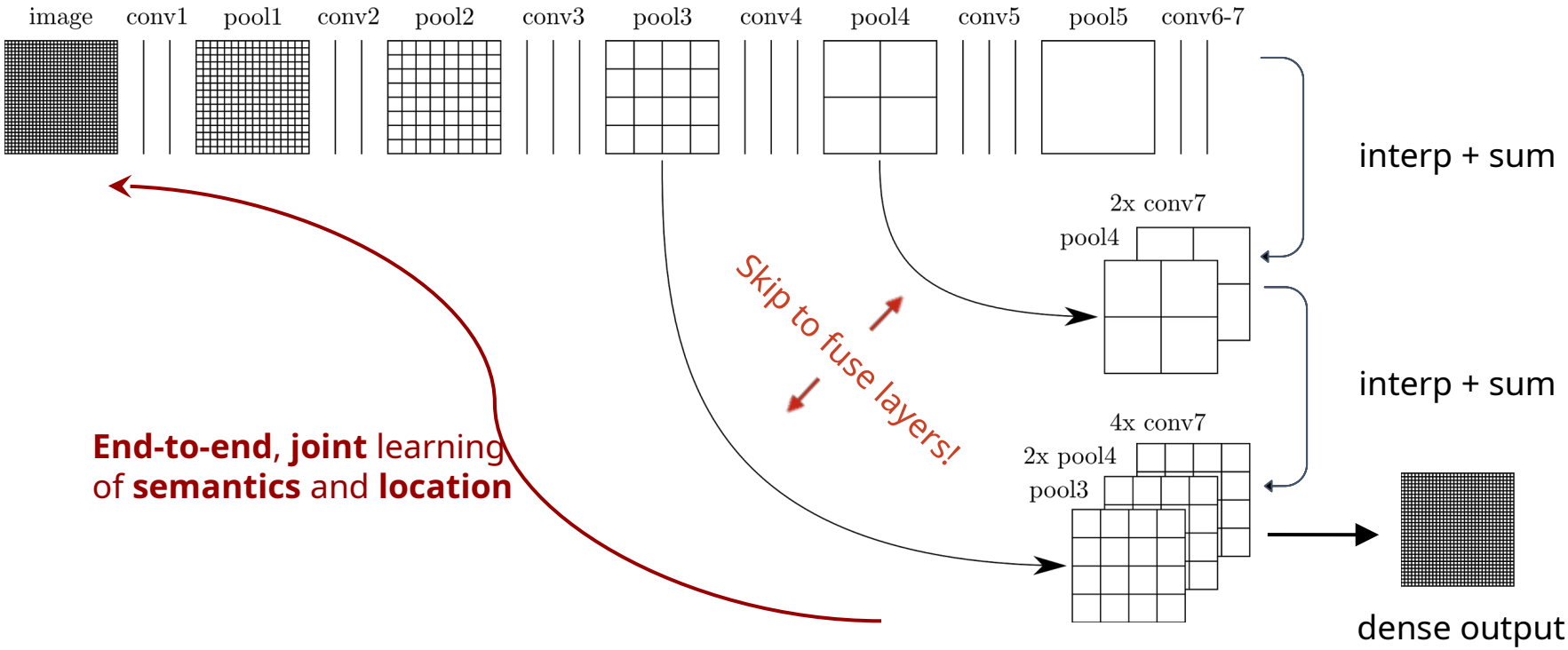
Or…
perform explicit bilinear upsampling before transpose convolution; let kernels of transpose convolution learn to fill in only high-frequency detail.

https://distill.pub/2016/deconv-checkerboard/

BUT…

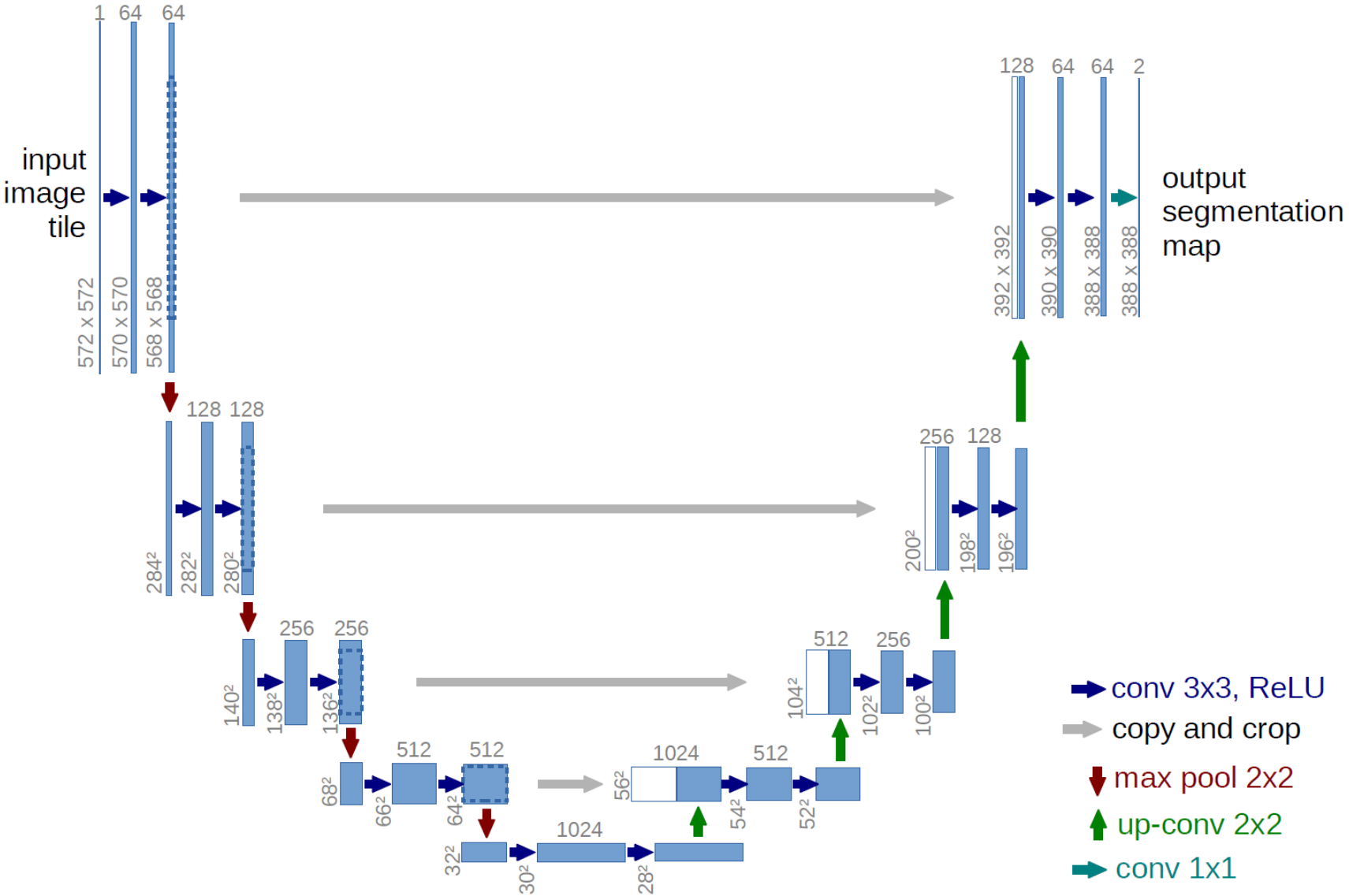
How do we ‘learn to create’ or ‘learn to restore’ new high frequency detail?

Learning upsampling kernels with skip layer refinement

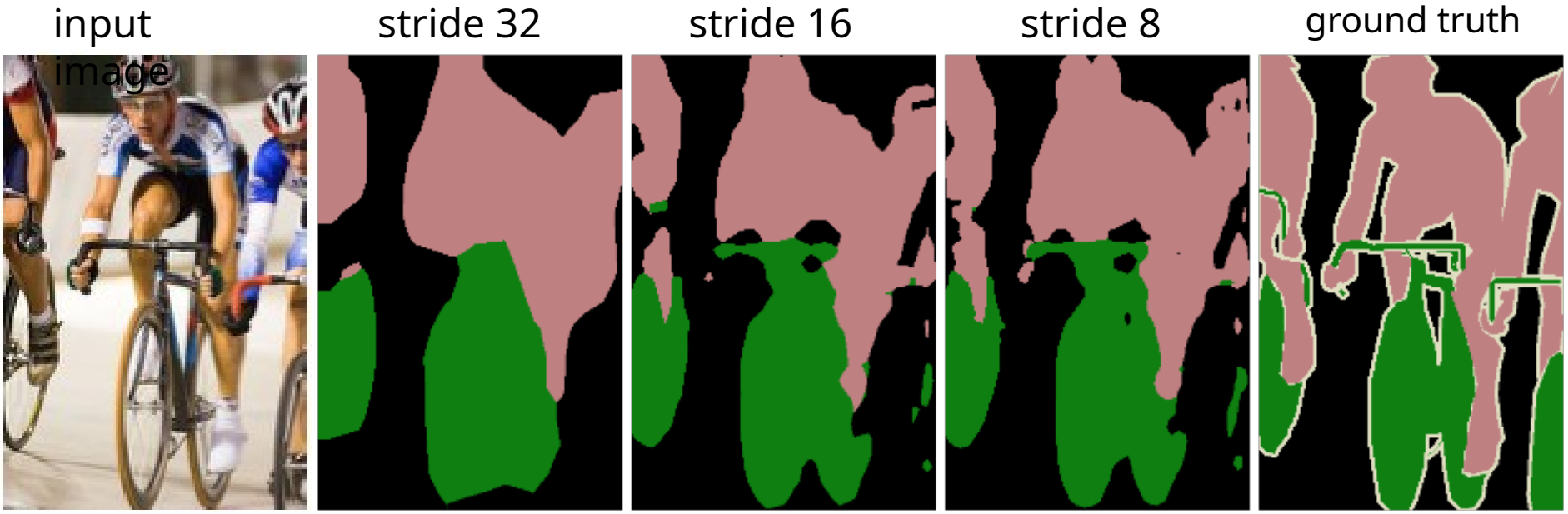


[Long et al.]

UNet [Ronneberger et al., 2015]



Skip layer refinement



[Long et al.]

Other usage example of an FCN revisiting im2gps

How much can an image tell about its geographic location?

6 million geo-tagged Flickr images

http://graphics.cs.cmu.edu/projects/im2gps/

im2gps (Hays & Efros, CVPR 2008)

Nearest Neighbors according to gist + bag of SIFT + color histogram + a few others


Paris


Paris


Paris


Paris


Paris


Paris


Paris


Madrid


Rome


Paris


Cuba

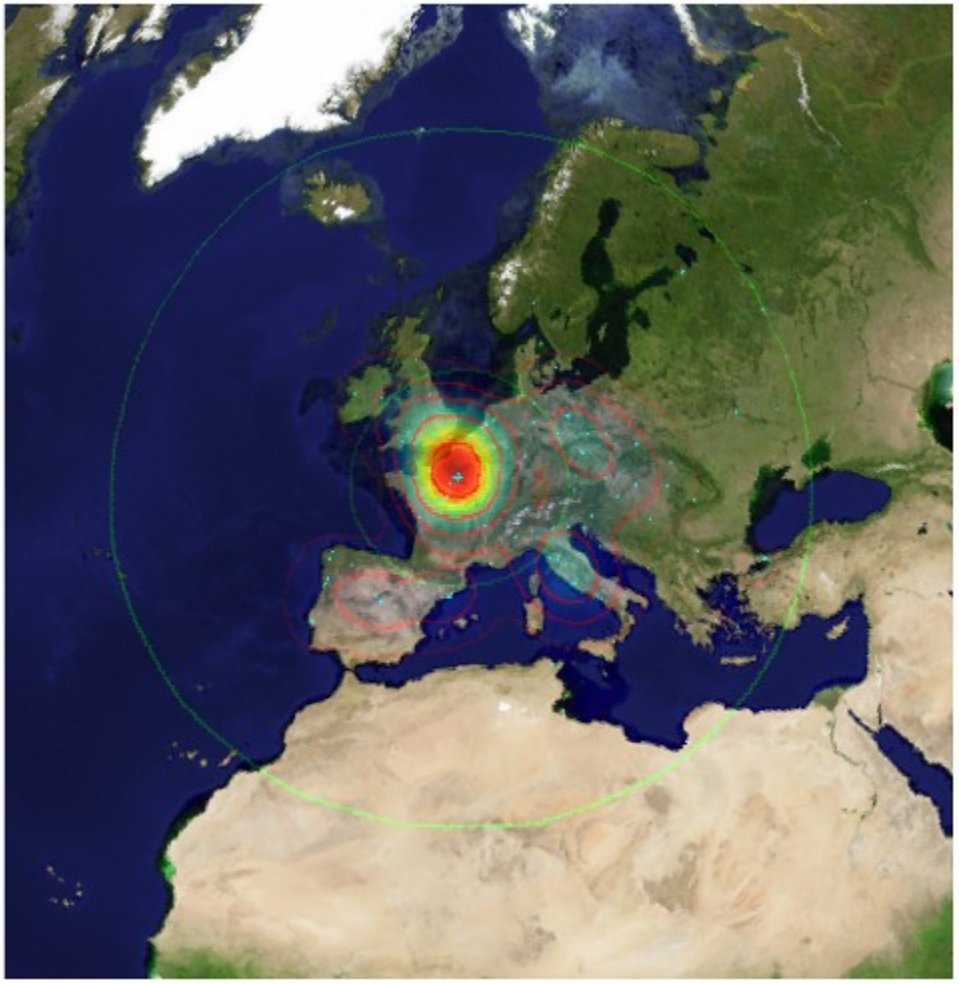

Paris


Paris


Poland


Paris


Paris



PlaNet - Photo Geolocation with Convolutional Neural Networks

Tobias Weyand, Ilya Kostrikov, James Philbin

ECCV 2016

Discretization of Globe

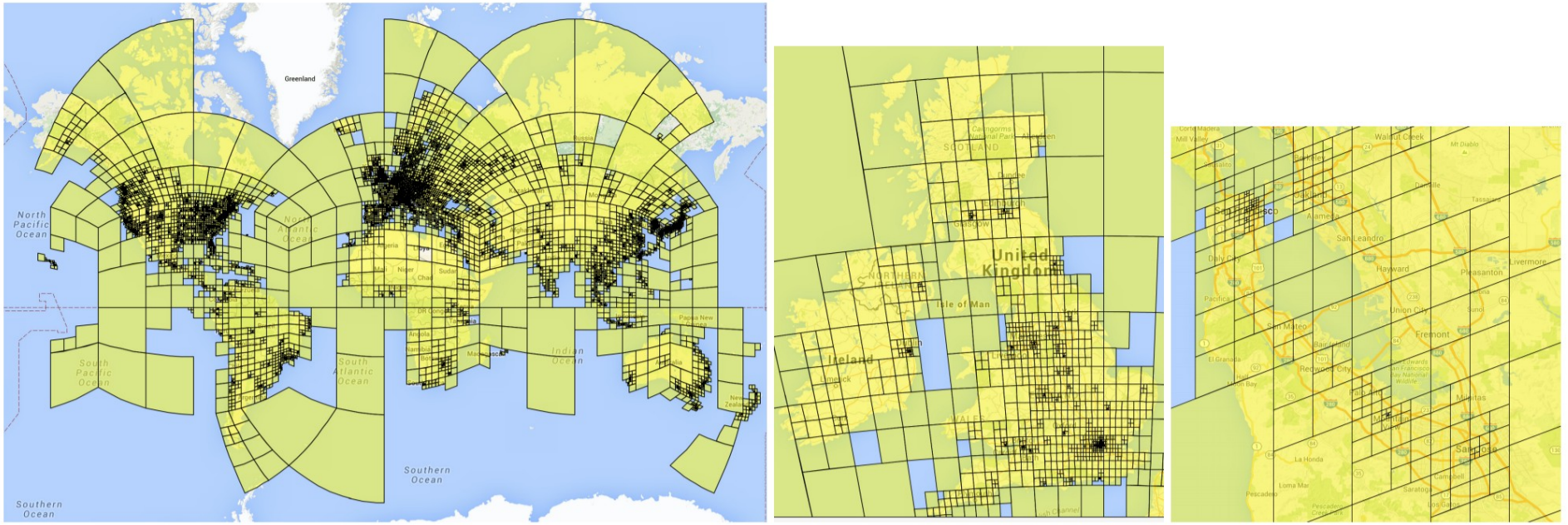


Figure 2. Left: Adaptive partitioning of the world into 26,263 S2 cells. Right: Detail views of Great Britain and Ireland and the San

Network and Training

- Network Architecture: Inception with 97M parameters
- 26,263 “categories” – places in the world

- 126 Million Web photos
- 2.5 months of training on 200 CPU cores

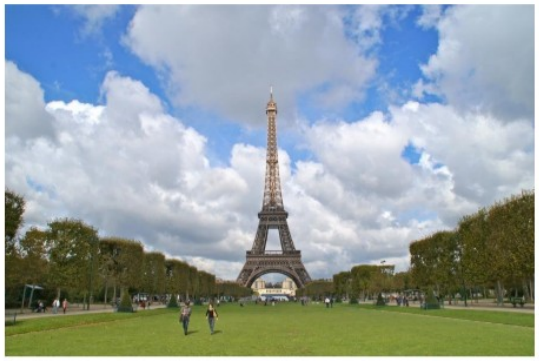
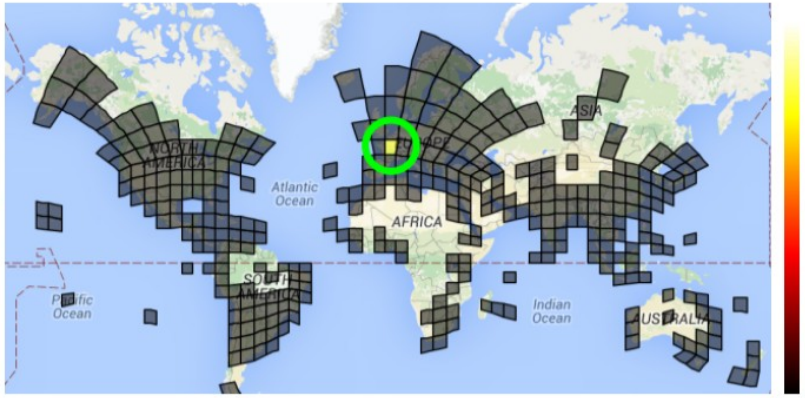
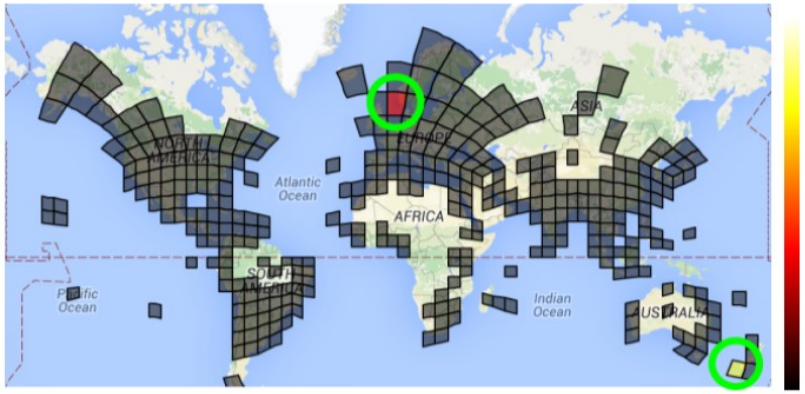


Photo CC-BY-NC by stevekc



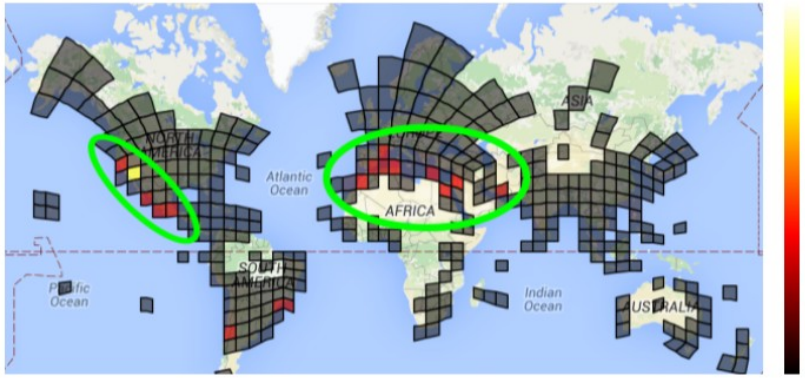
(a)

Photo CC-BY-NC by edwin.11



(b)

Photo CC-BY-NC by jonathanfh



Namibia / Botswana

Photo by jamie.lovelock / CC BY NC

Photo by MongoosePhotography / CC BY NC

Photo by Mister-E / CC BY NC

Photo by dalangalma / CC BY NC

Photo by siamjack / CC BY NC

Kauai, Hawaii

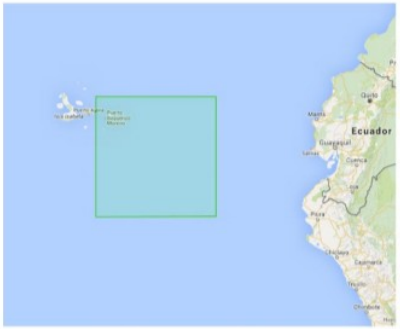
Photo by ryan + sarah / CC BY NC

Photo by stuartlchambers / CC BY NC

Photo by samgrover / CC BY NC

Photo by steuben / CC BY NC

Photo by steve-stevens / CC BY NC



Galapagos Islands

Photo by p.j.k. / CC BY NC

Photo by victor408 / CC BY NC

Photo by Domen Jakus / CC BY NC

Photo by cvanholder / CC BY NC

Photo by rwoan / CC BY NC

Paris

Photo by feliven / CC BY NC

Photo by fred_v / CC BY NC

Photo by Turansa Tours / CC BY NC

Photo by JA_F5 / CC BY NC

Photo by CedEm photographies / CC BY NC

PlaNet (2016) vs im2gps (2008,2009)

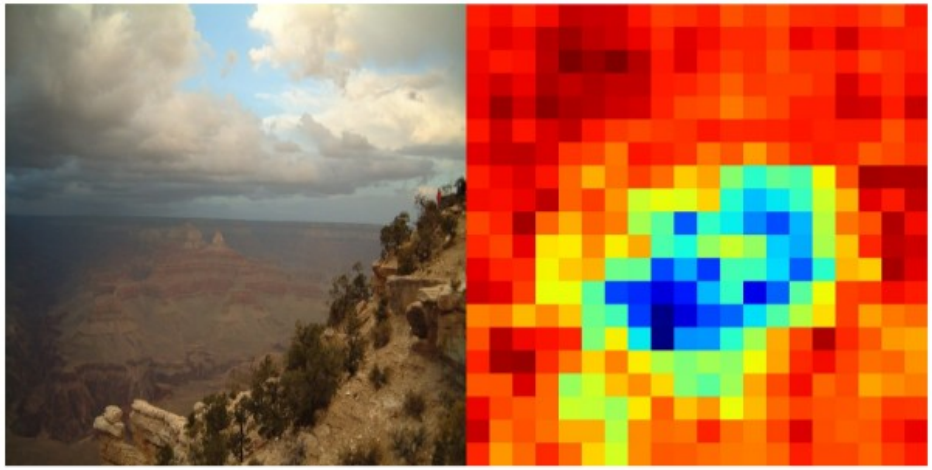
Percentages are the fraction of images from the Im2GPS test set that were localized within the given radius.

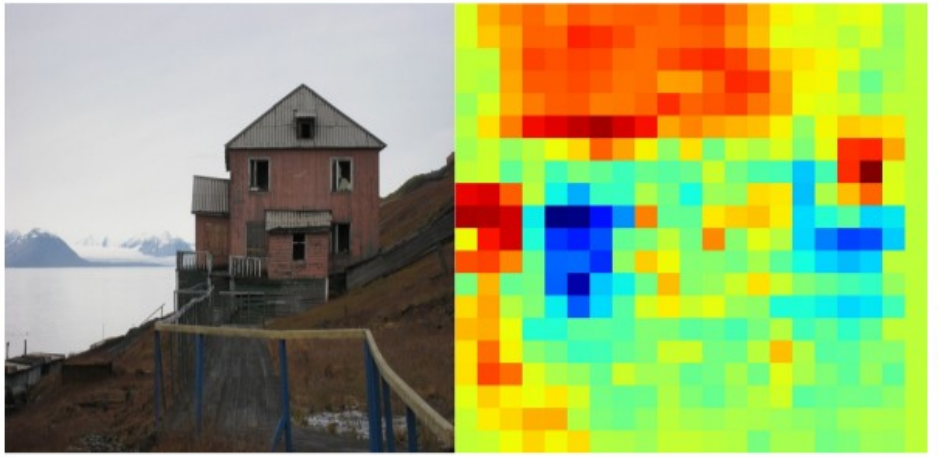
Method	Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS (orig) [17]		12.0%	15.0%	23.0%	47.0%
Im2GPS (new) [18]	2.5%	21.9%	32.1%	35.4%	51.9%
PlaNet	**8.4%**	**24.5%**	**37.6%**	**53.6%**	**71.3%**

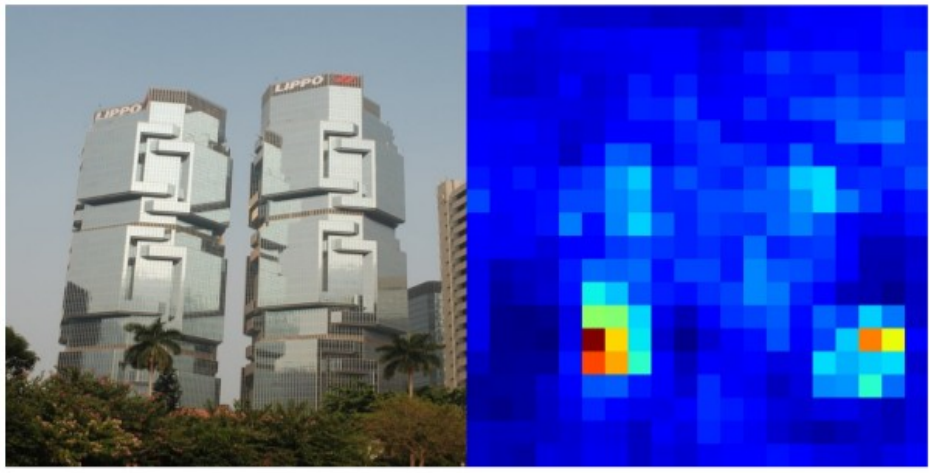
Median localization error (km) by image category.

Method	Manmade Landmark	Natural Landmark	City Scene	Natural Scene	Animal
Im2GPS (new)	61.1	37.4	3375.3	5701.3	6528.0
PlaNet	74.5	61.0	212.6	1803.3	1400.0

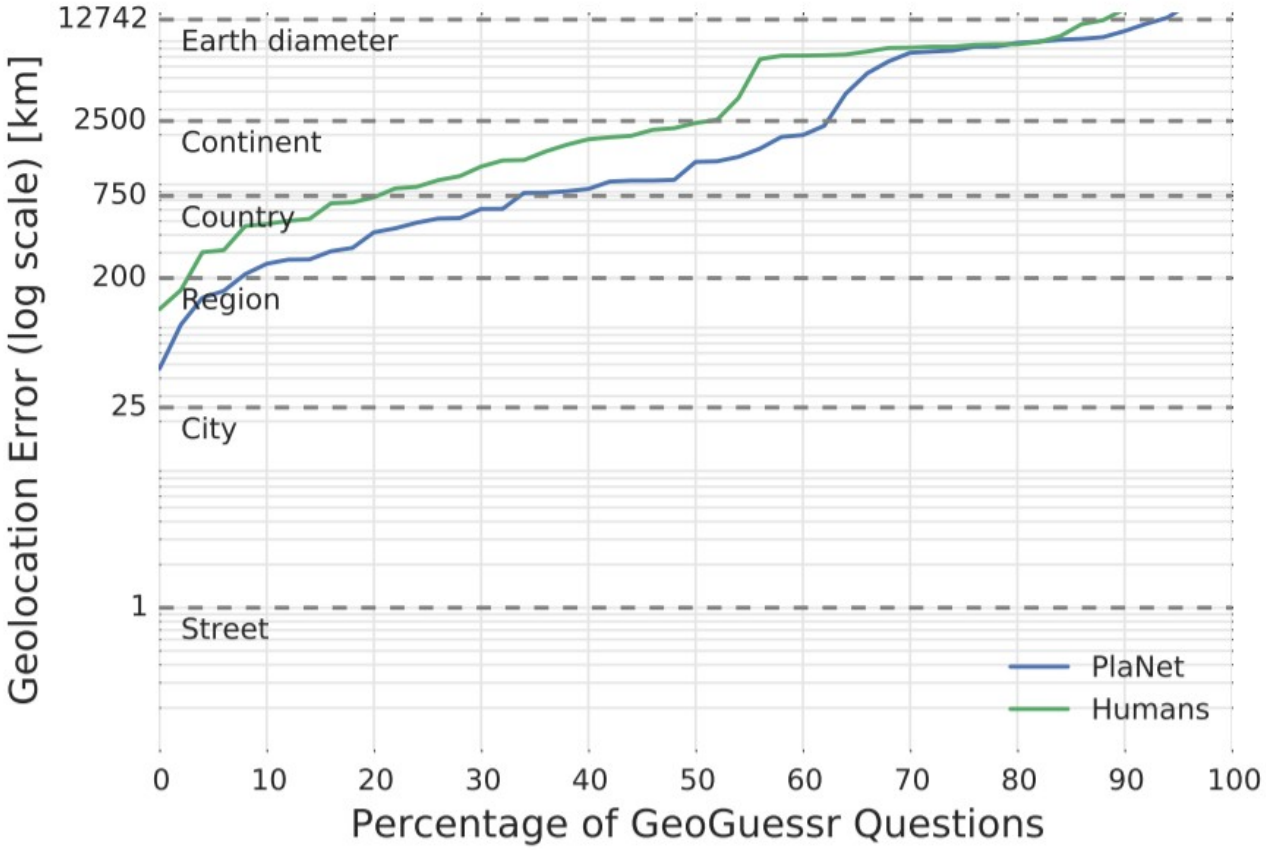
Spatial support for decision







PlaNet vs. Humans



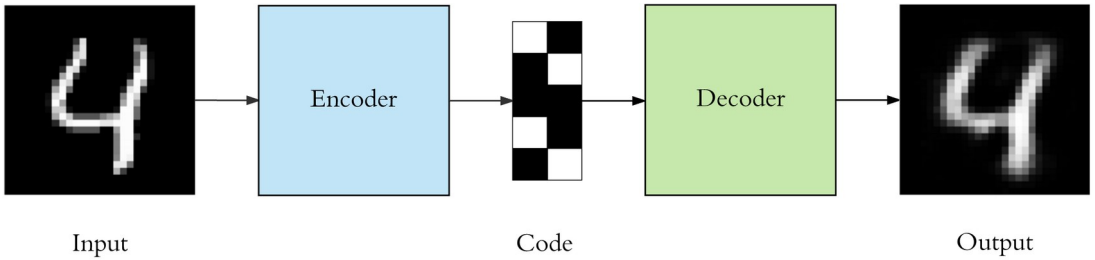
PlaNet vs Humans

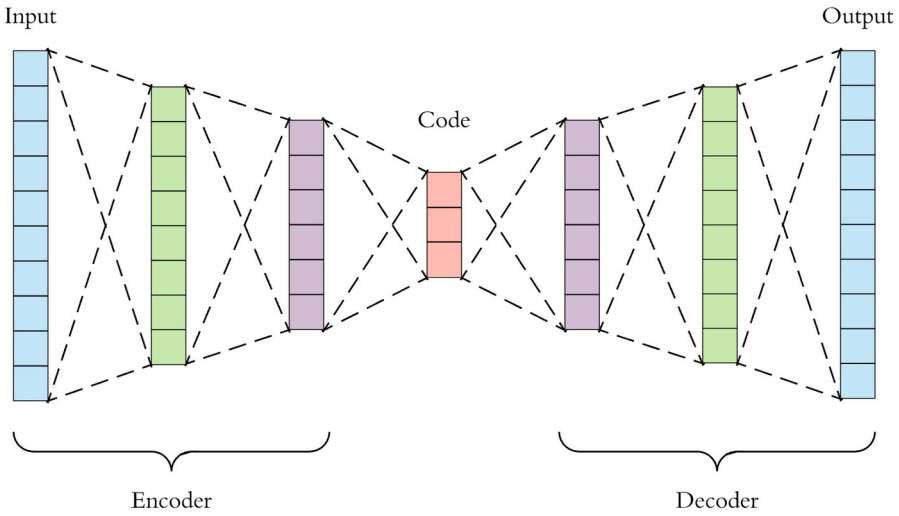






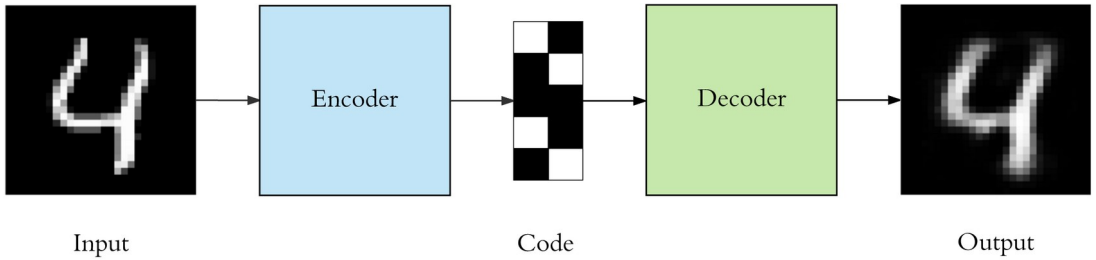

Autoencoders

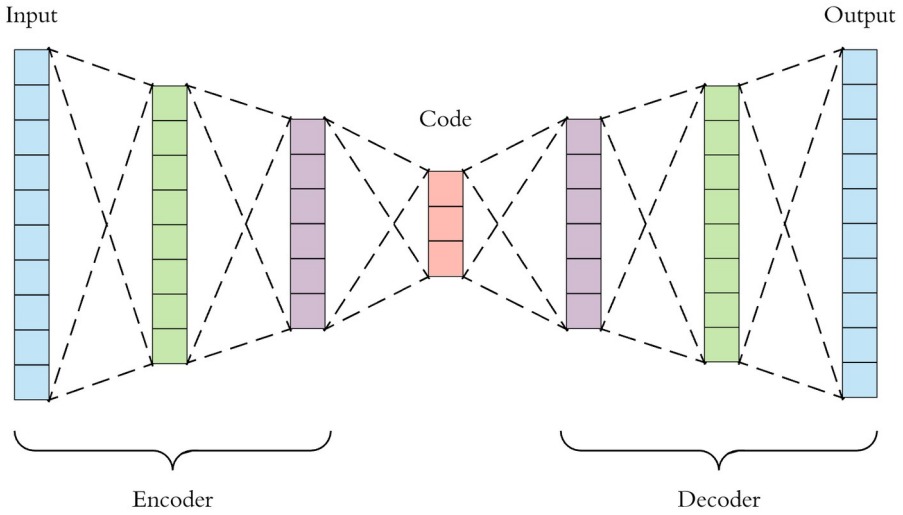


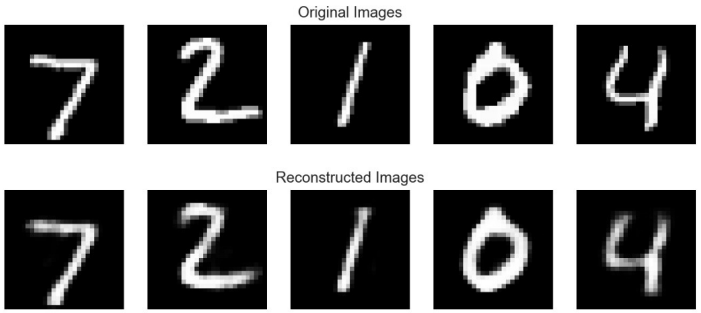


https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Autoencoders

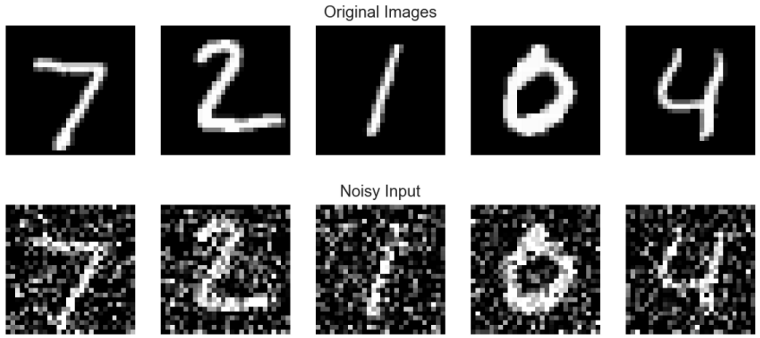


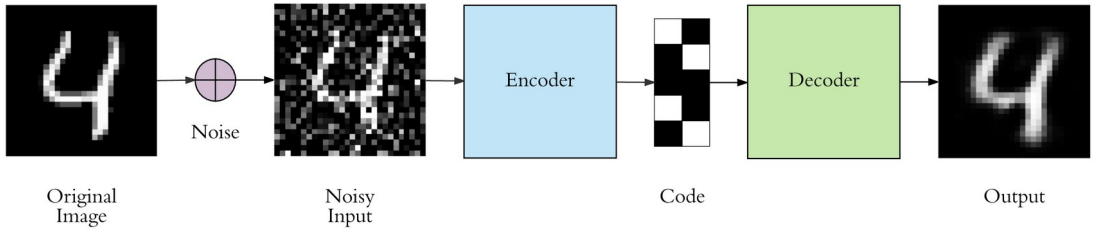


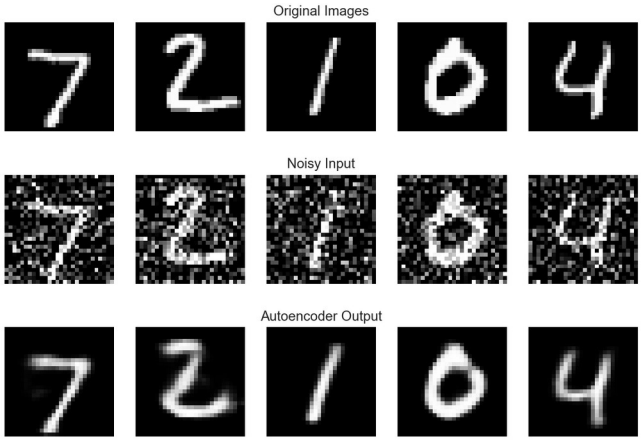


https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Autoencoders

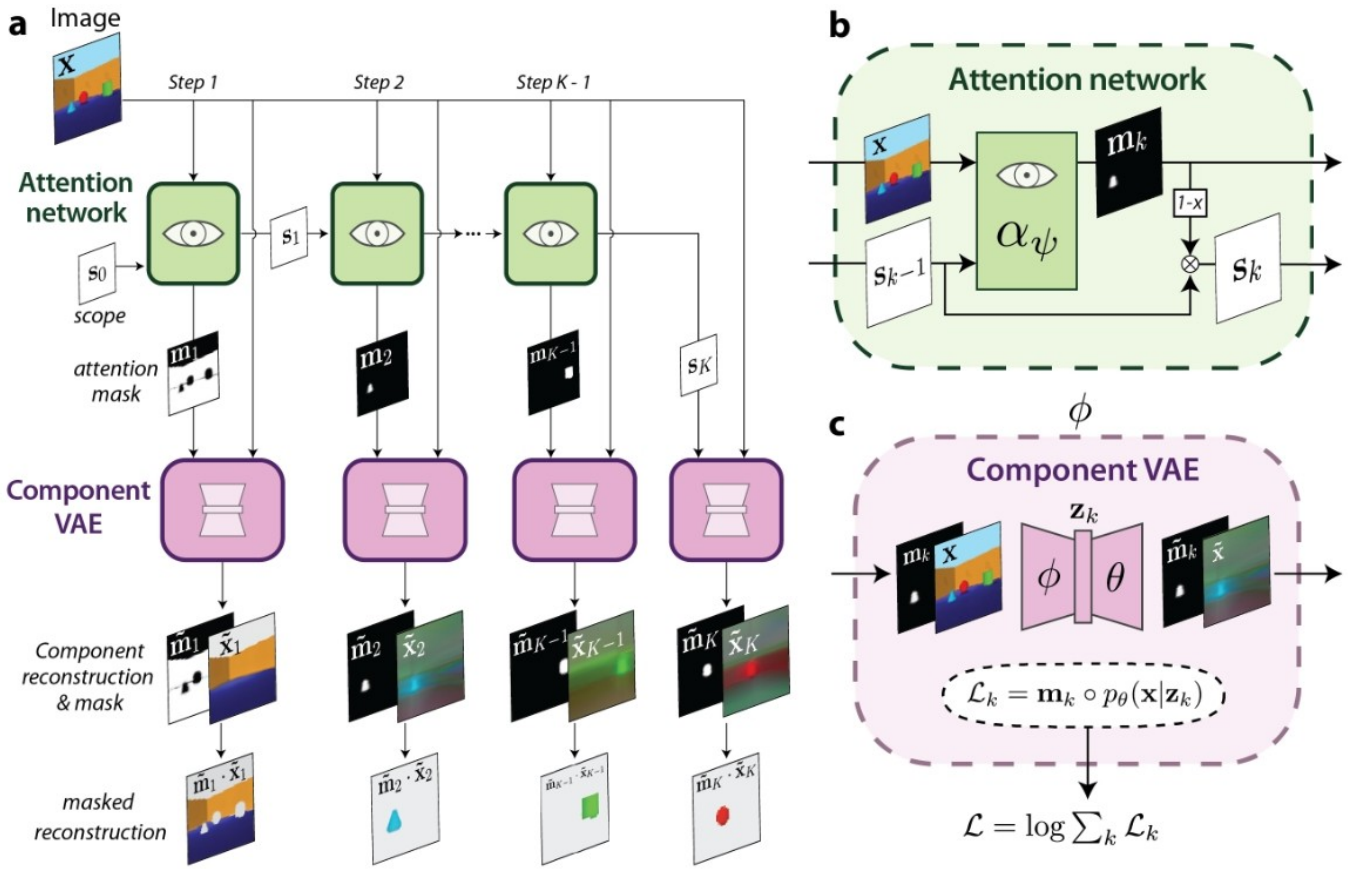






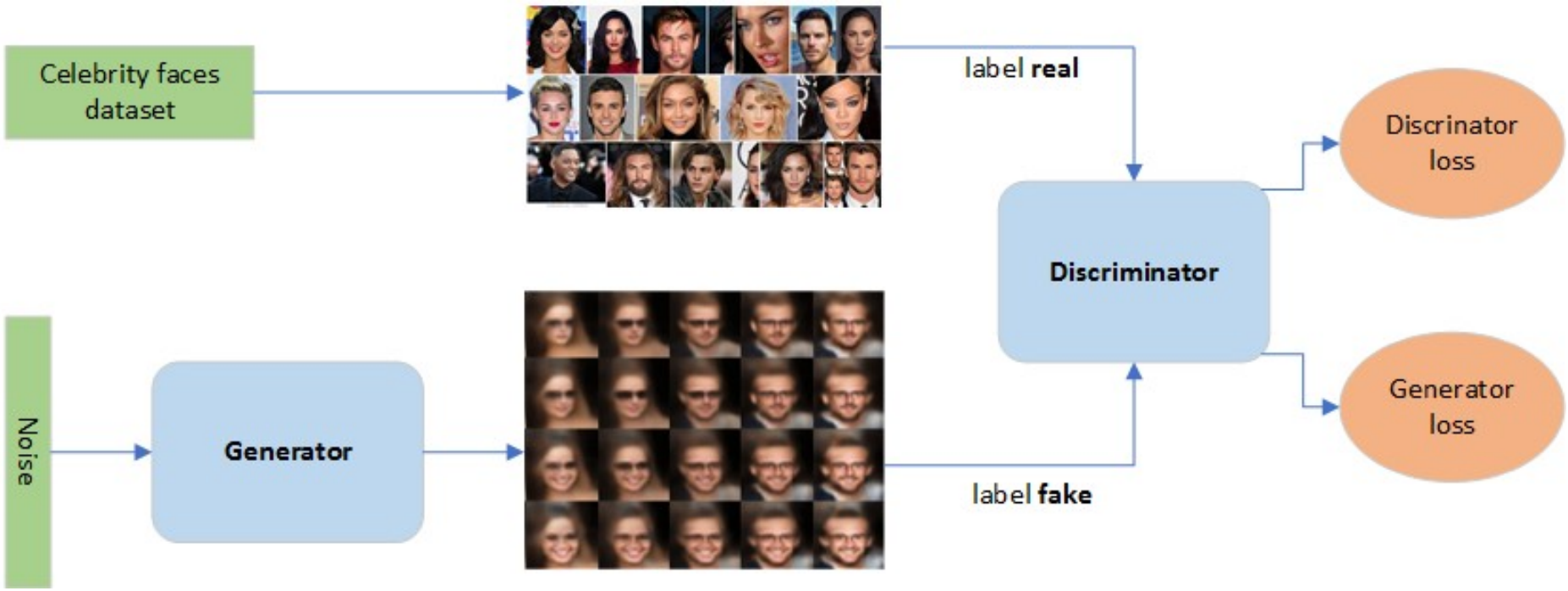
https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Autoencoders - MOnet



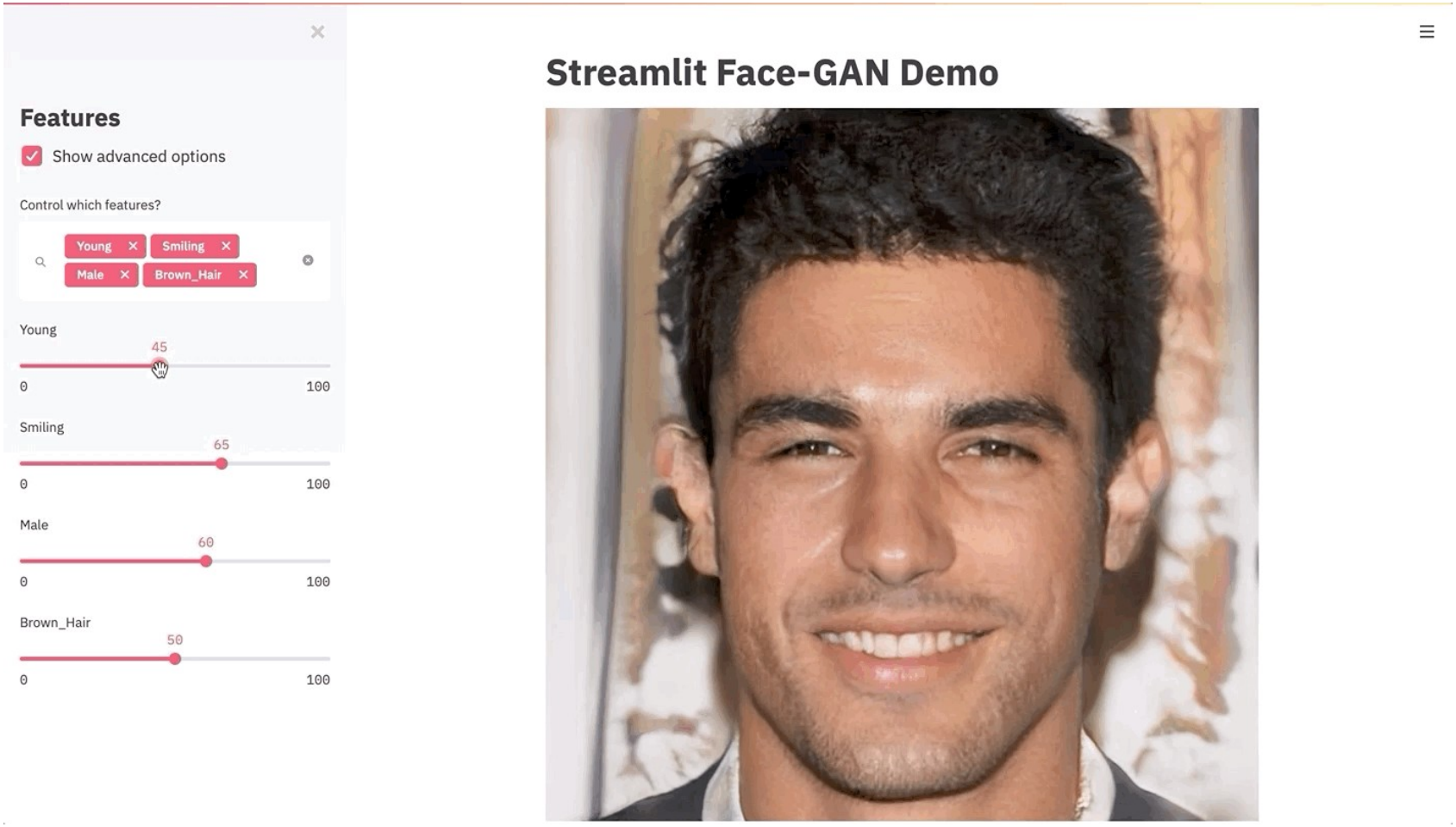
(Burgess et al., 2019)

Generative Adversarial Networks



https://towardsdatascience.com/generative-adversarial-network-gan-for-dummies-a-step-by-step-tutorial-fdefff170391

Generative Adversarial Networks



https://towardsdatascience.com/build-an-app-to-synthesize-photorealistic-faces-using-tensorflow-and-streamlit-dd2545828021

Generative Models - Example

(Siarohin et al., 2019)

Some refs

Intro: https://cnl.salk.edu/~schraudo/teach/NNcourse/intro.html

Deep learning rules of thumb (Jeff Macaluso) summary of Goodfellow et al book:
https://jeffmacaluso.github.io/post/DeepLearningRulesOfThumb/

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

Recipe for training nn: http://karpathy.github.io/2019/04/25/recipe/

Useful basic guidelines: https://yerevann.com/a-guide-to-deep-learning/

Lots of interesting stuff out there for example on:

generalization of networks: https://singularityhub.com/2019/10/03/deep-learning-networks-cant-generalize-but-theyre-learning-from-the-brain/

top 10 architectures: https://medium.com/cracking-the-data-science-interview/a-gentle-introduction-to-neural-networks-for-machine-learning-d5f3f8987786

backprop alternatives: https://medium.com/@sallyrobotics.blog/backpropagation-and-its-alternatives-c09d306aae4c

how deep should we go: https://link.springer.com/article/10.1007/s12559-019-09667-7