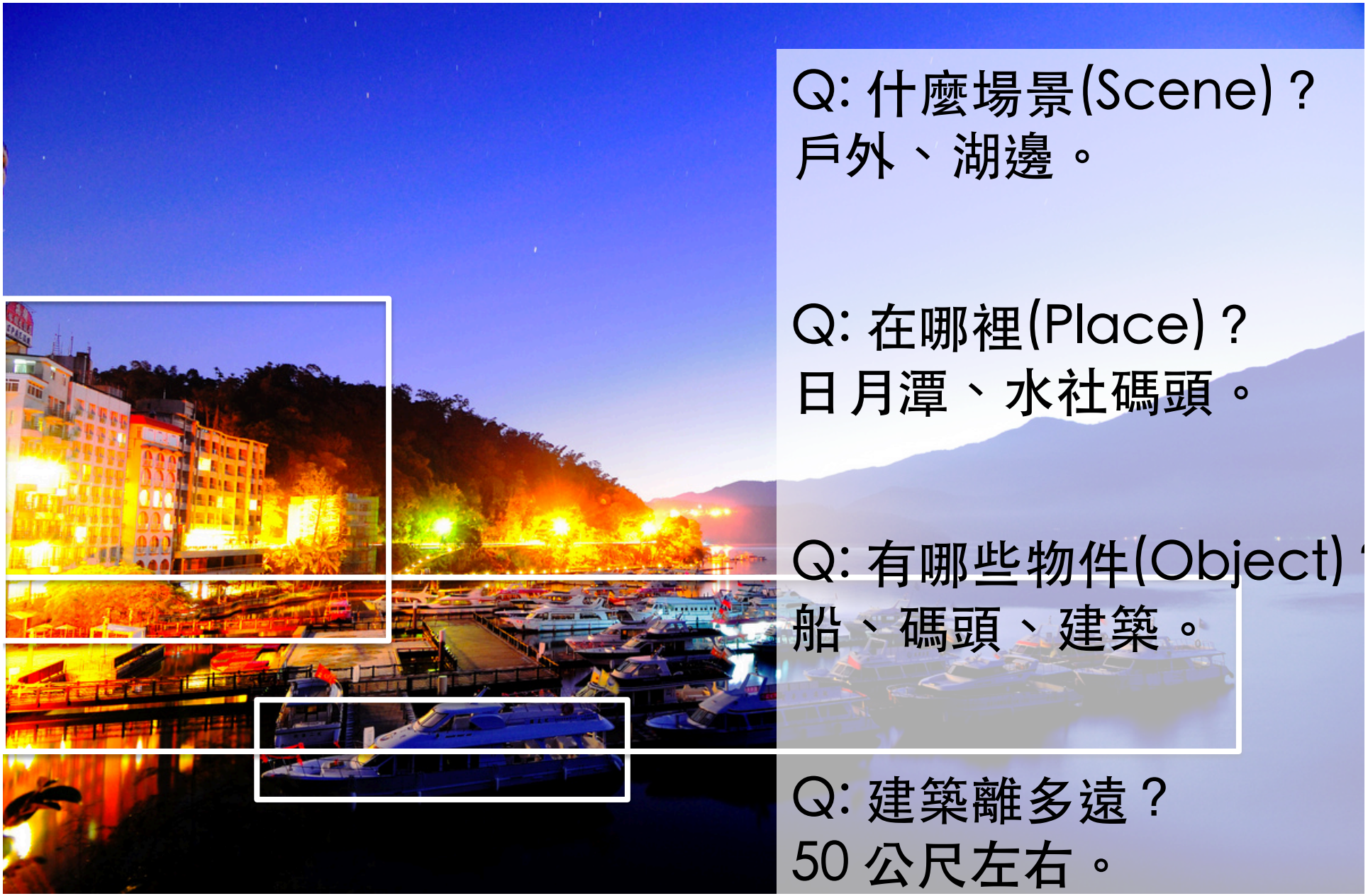
Deep Learning in Computer Vision – Convolutional Neural Network

國立清華大學

孫民教授

VSLab

電腦視覺 Computer Vision



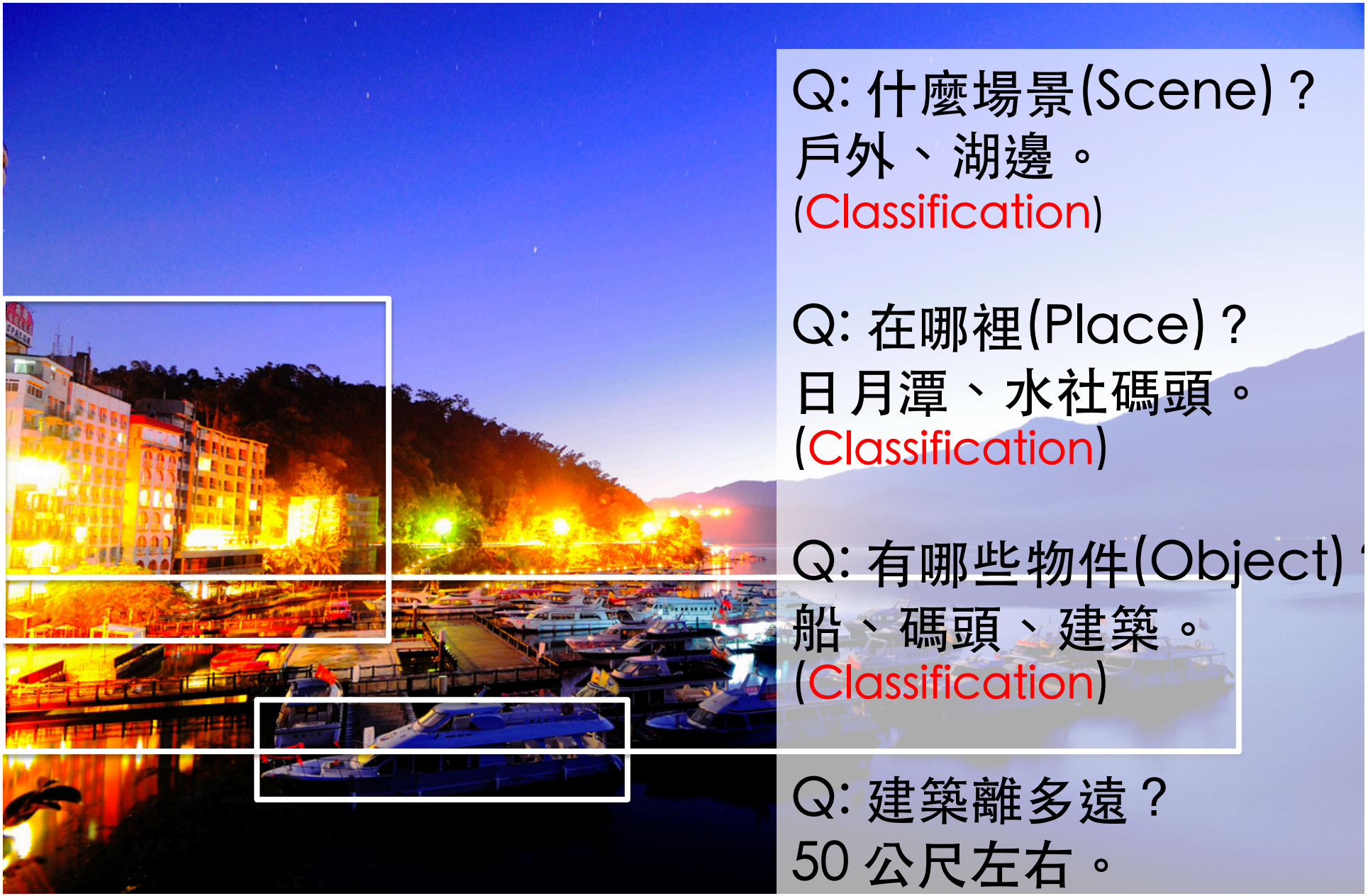
Q: 什麼場景(Scene)？
戶外、湖邊。

Q: 在哪裡(Place)？
日月潭、水社碼頭。

Q: 有哪些物件(Object)？
船、碼頭、建築。

Q: 建築離多遠？
50 公尺左右。

電腦視覺 Computer Vision



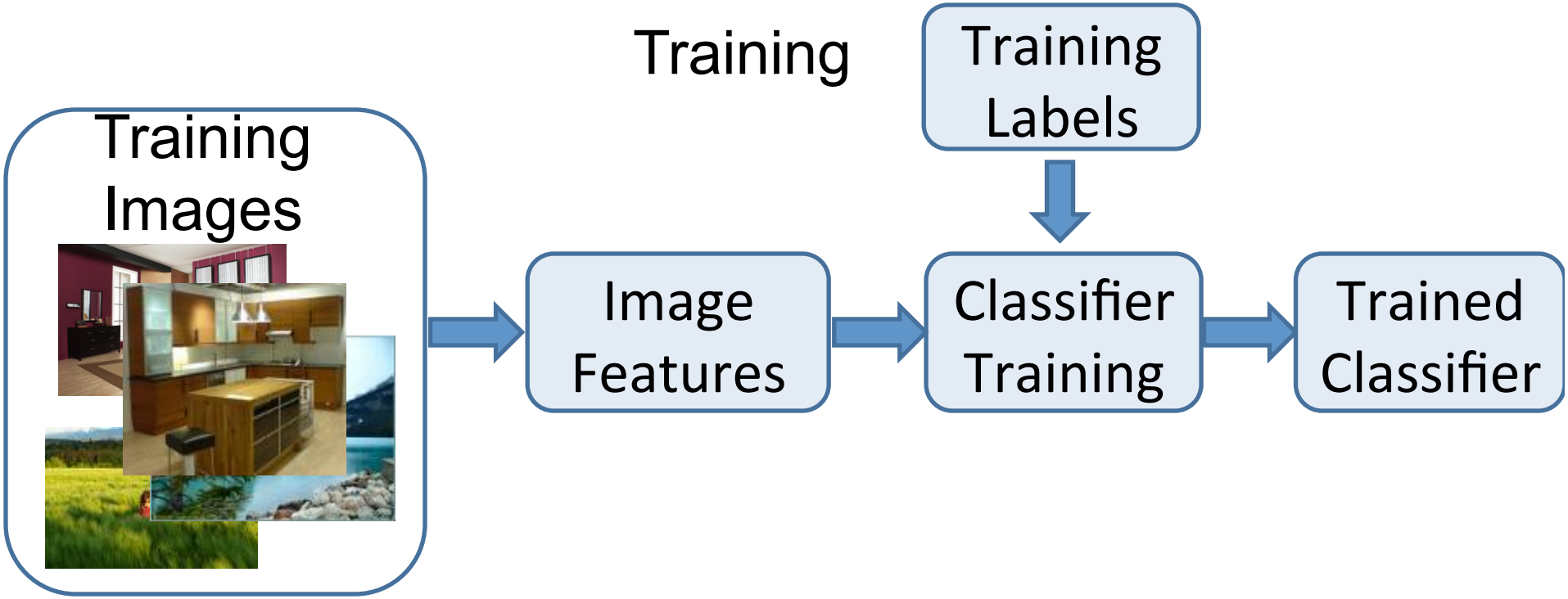
Q: 什麼場景(Scene)？
戶外、湖邊。
(Classification)

Q: 在哪裡(Place)？
日月潭、水社碼頭。
(Classification)

Q: 有哪些物件(Object)？
船、碼頭、建築。
(Classification)

Q: 建築離多遠？
50 公尺左右。

Image Classification



Slides: Jame Hays, Isabelle Guyon, Erik Sudderth, Mark Johnson, Derek Hoiem

Image Classification

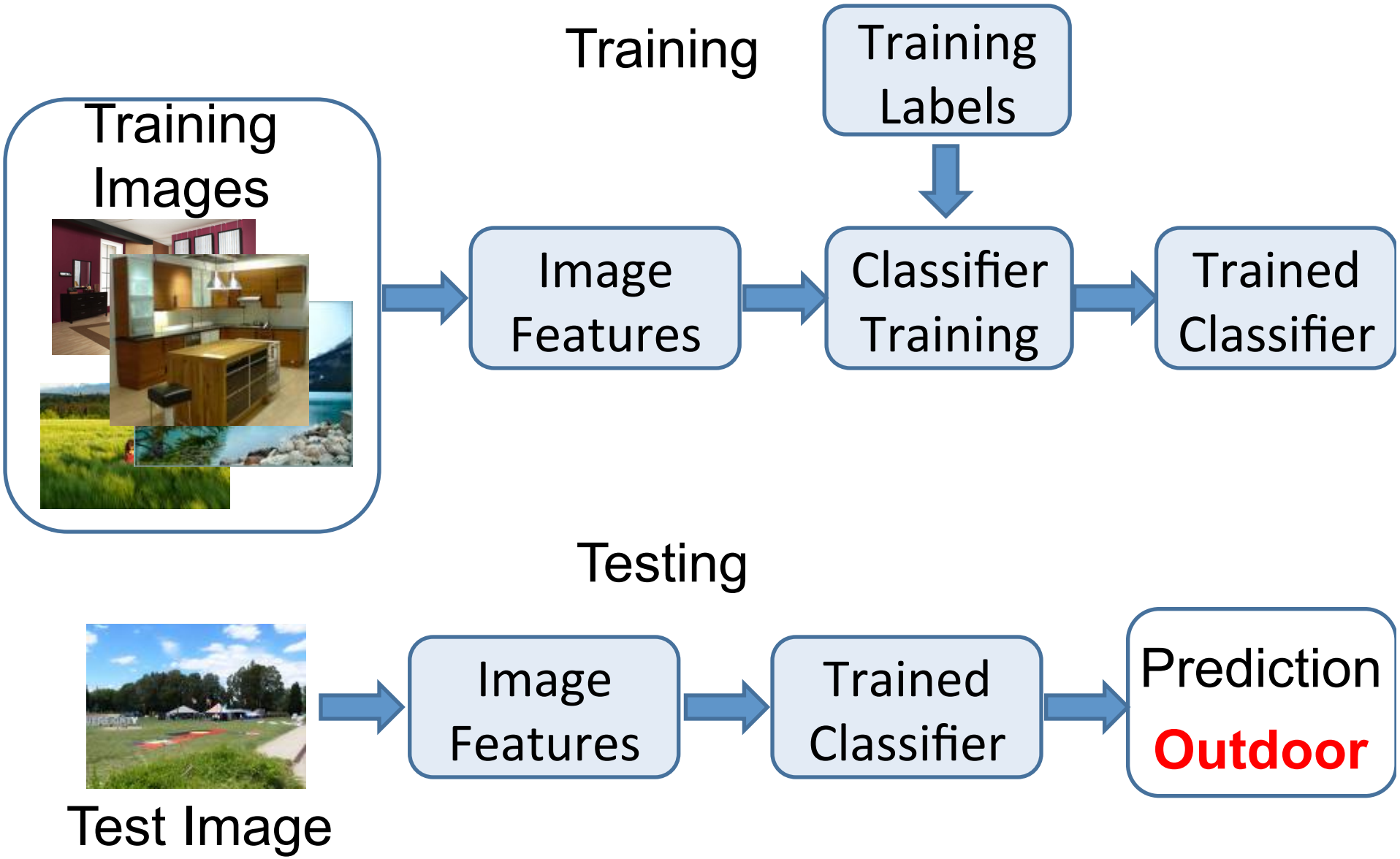
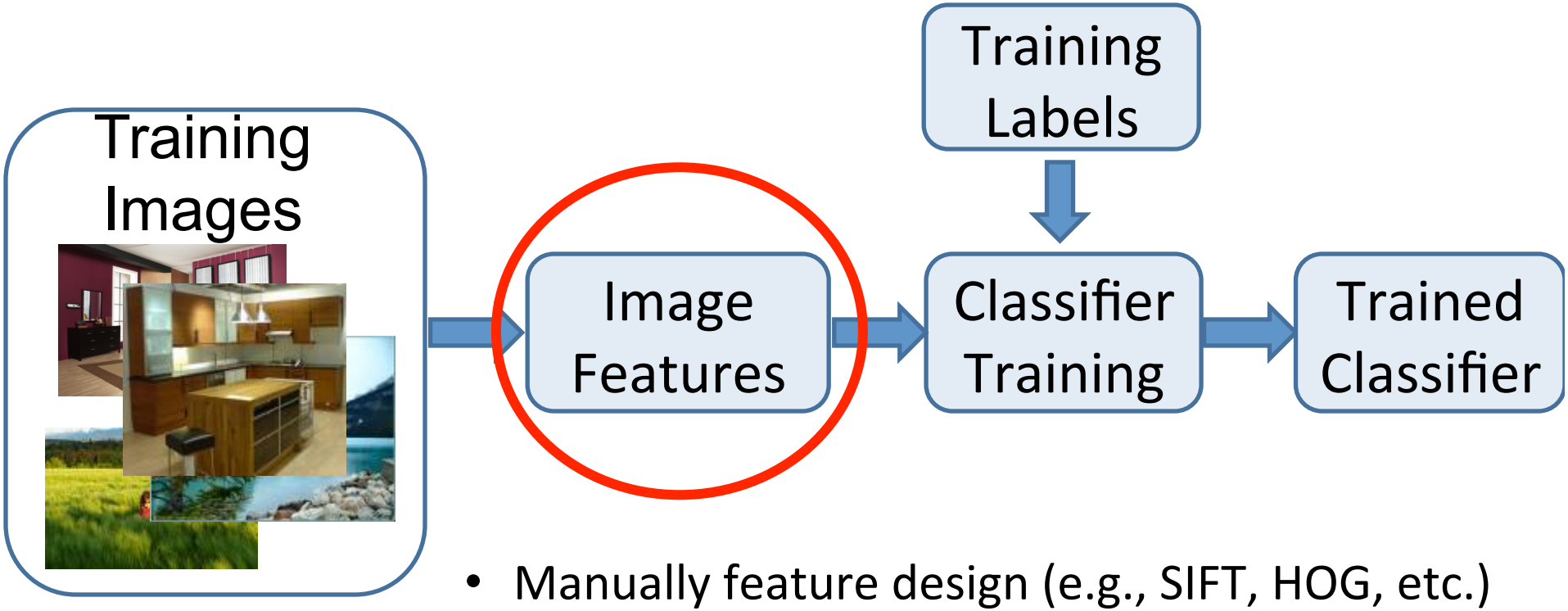


Image Classification

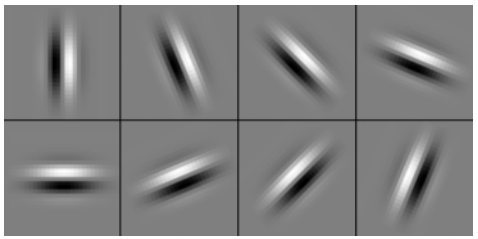


- Manually feature design (e.g., SIFT, HOG, etc.)
- Multiple separately learned layers
  (e.g., K-means, Pyramid, etc.)

SIFT Descriptor

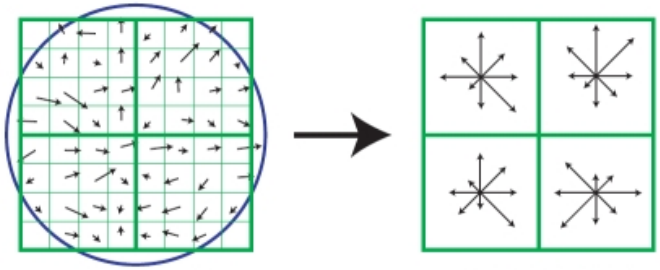
Image Pixels

Apply oriented filters

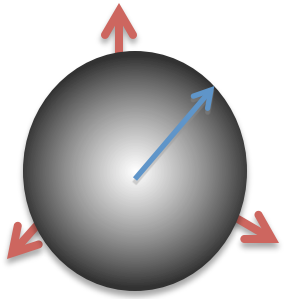


Lowe [IJCV 2004]

Spatial pool (Sum)



Normalize to unit length



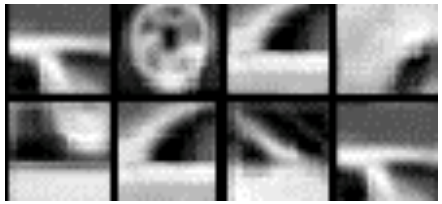
Feature Vector

slide credit: R. Fergus

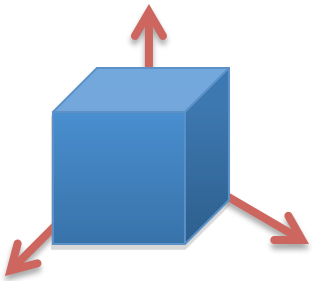
Spatial Pyramid Matching

SIFT Features

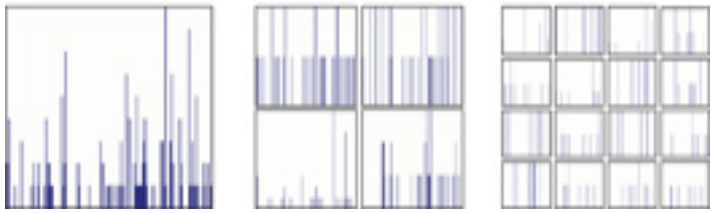
Filter with Visual Words



Max



Multi-scale spatial pool (Sum)



Classifier

Lazebnik, Schmid, Ponce [CVPR 2006]

slide credit: R. Fergus

Comparison

- Classical CV

Hand-crafted Feature Extractor → "Simple" Trainable Classifier

- Deep Learning

Trainable Feature Extractor → **Trainable** Classifier

End-to-End Learning

Slides: Y LeCun
MA Ranzato

SIFT HoG

K-means Sparse Coding

Pooling

Classifier

Fixed

Unsupervised

supervised

Low-level Features

Mid-level Features

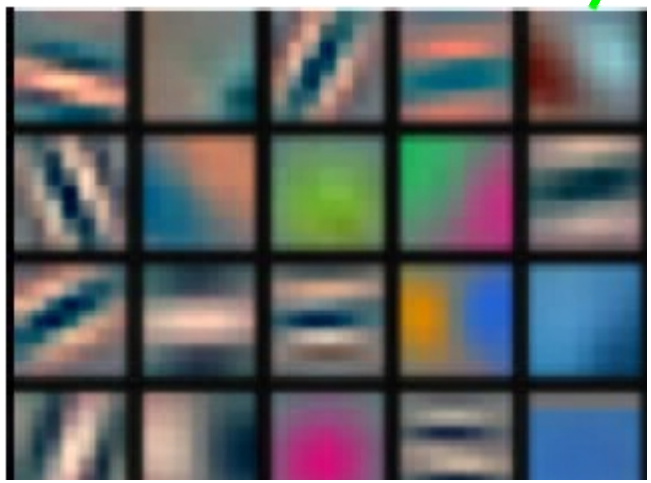
2012

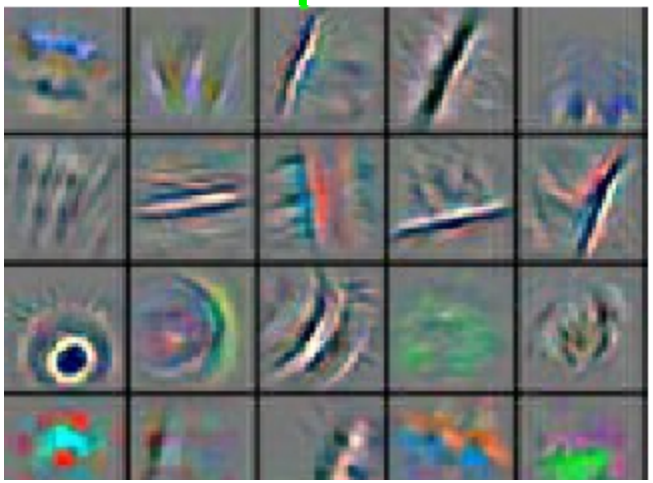
Low-Level Feature

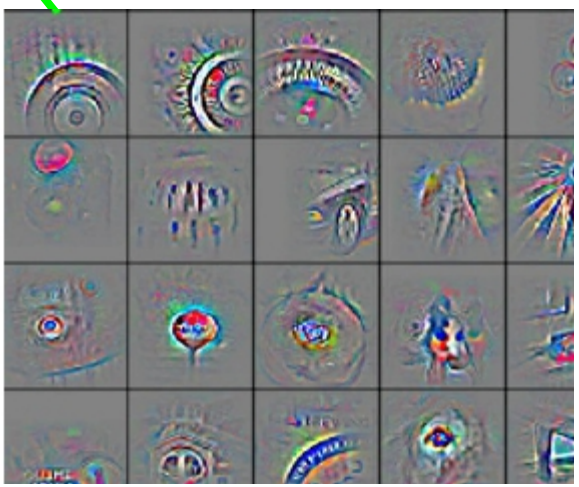
Mid-Level Feature

High-Level Feature

Trainable Classifier







1K Image Classification

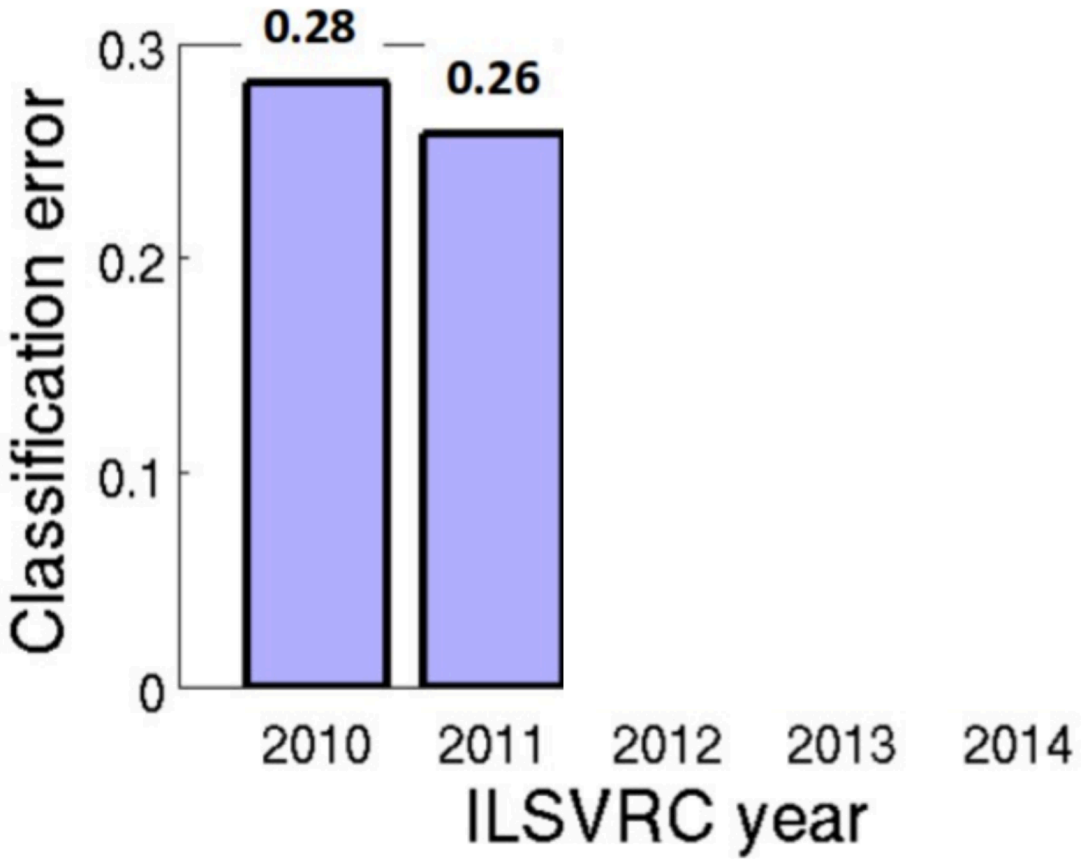


Figure from Olga Russakovsky ECCV'14 workshop

Deep Learning

1K Image Classification

Convolution Neuron Networks (CNN)

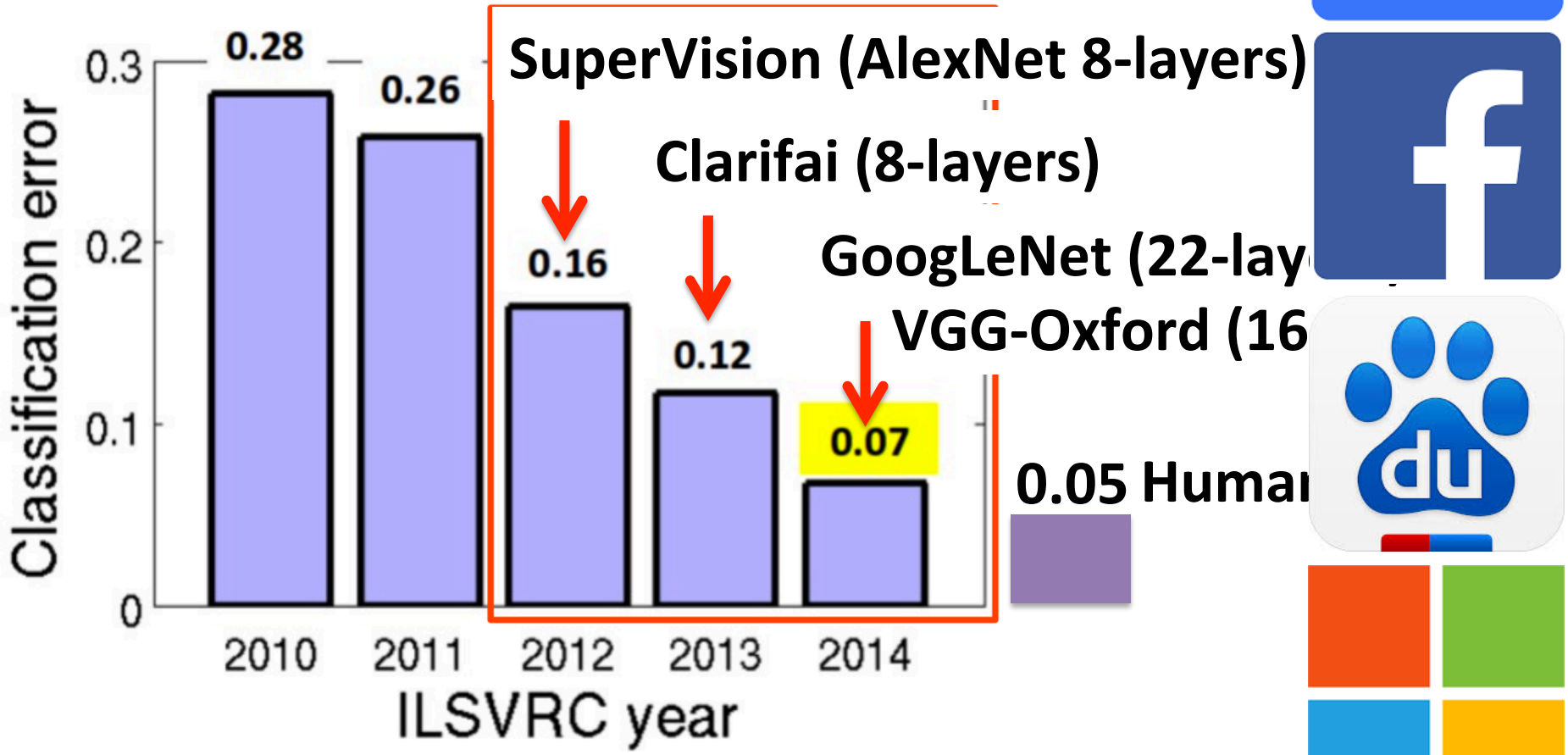
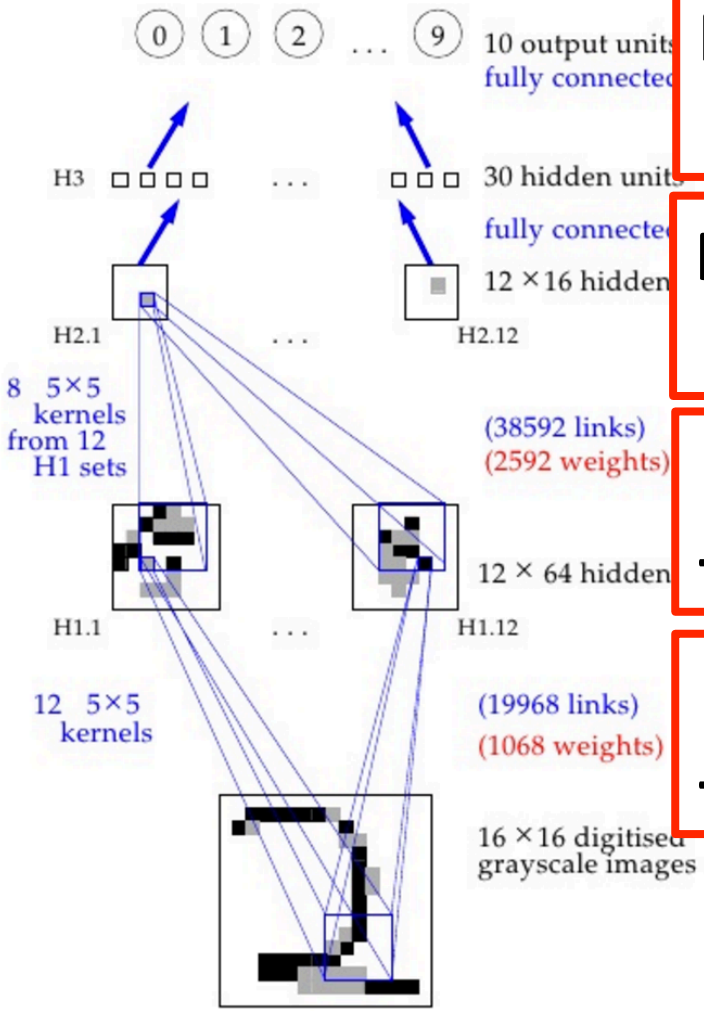


Figure from Olga Russakovsky ECCV'14 workshop

Classic CNN: LeNet



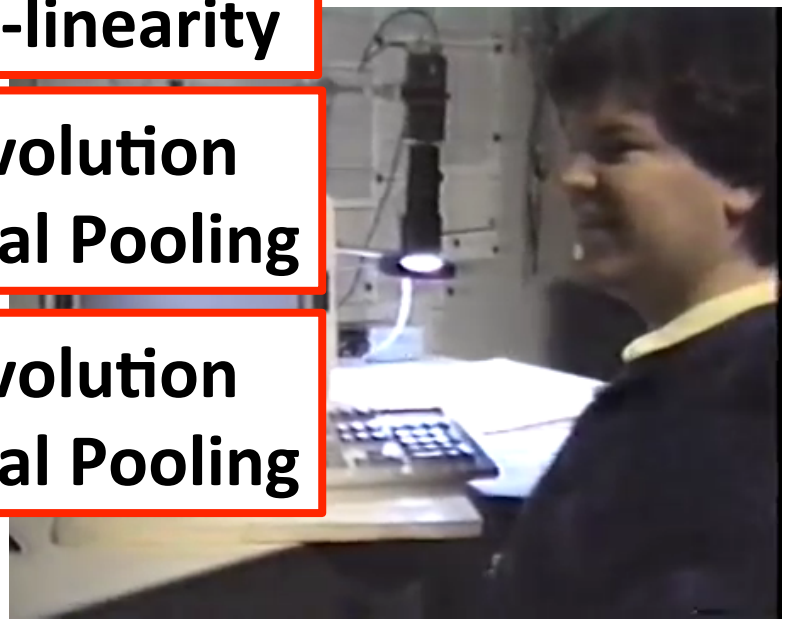
Before weight sharing 64660 links
After weight sharing 9760 weights

Fully Connected + Non-linearity

Fully Connected + Non-linearity

Convolution +Spatial Pooling

Convolution +Spatial Pooling



LeNet-1, 1993

Backpropagation Applied to Handwritten Zip Code. Yann Lecun, 1989

What is 2D Convolution?

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0

$h[.,.]$

	0								

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10							

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20						

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30					

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

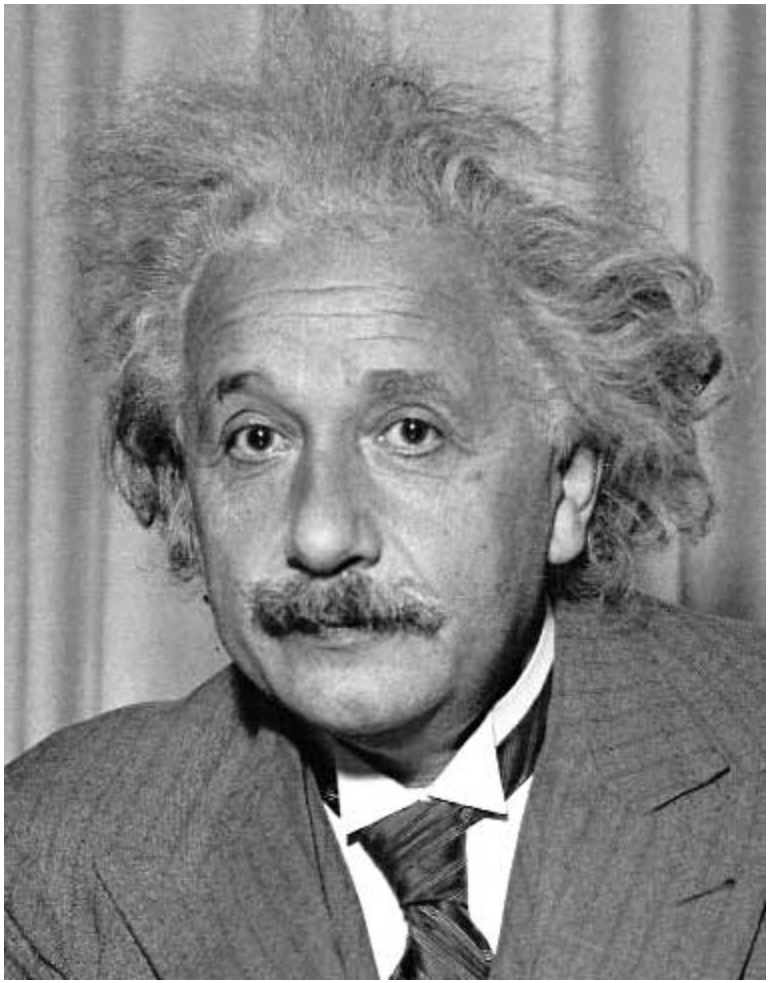
$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

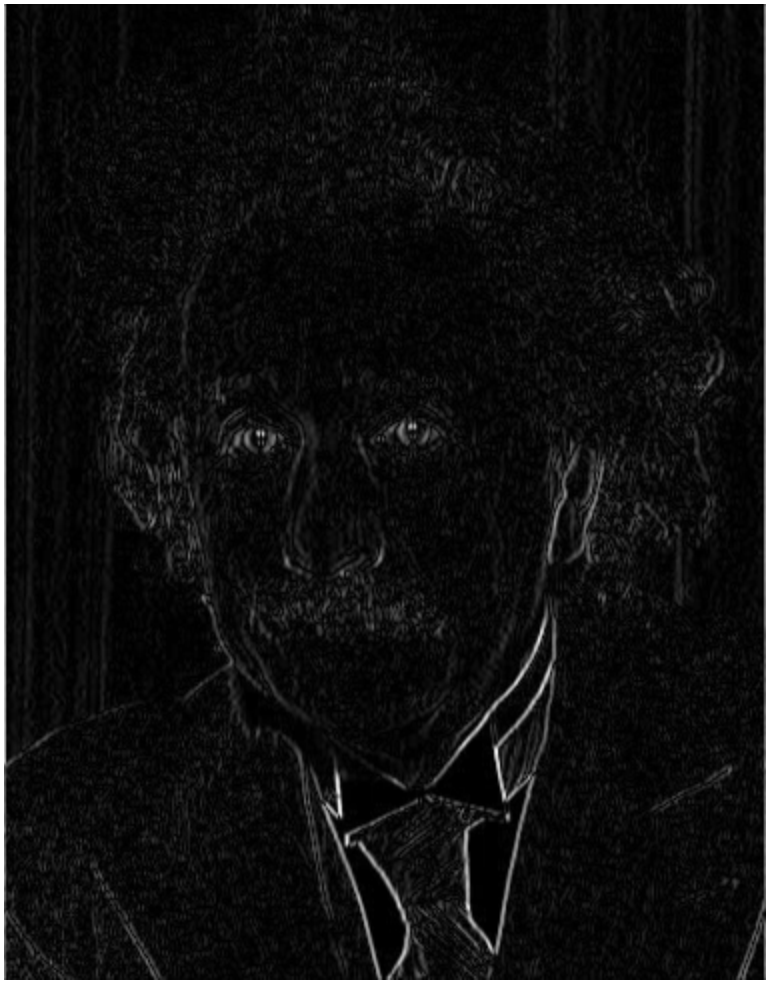
Credit: S. Seitz

Image filtering



1	0	-1
2	0	-2
1	0	-1

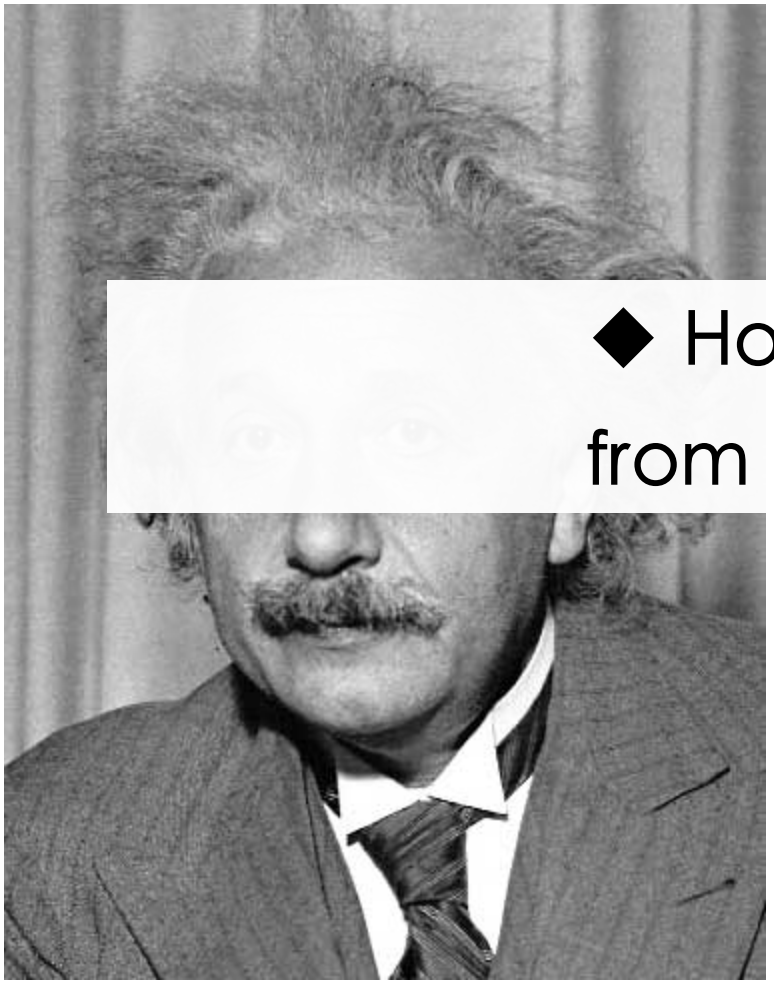
Sobel



Vertical Edge
(absolute value)

http://en.wikipedia.org/wiki/Sobel_operator

Image filtering



- How to learn filters from Big Visual data?

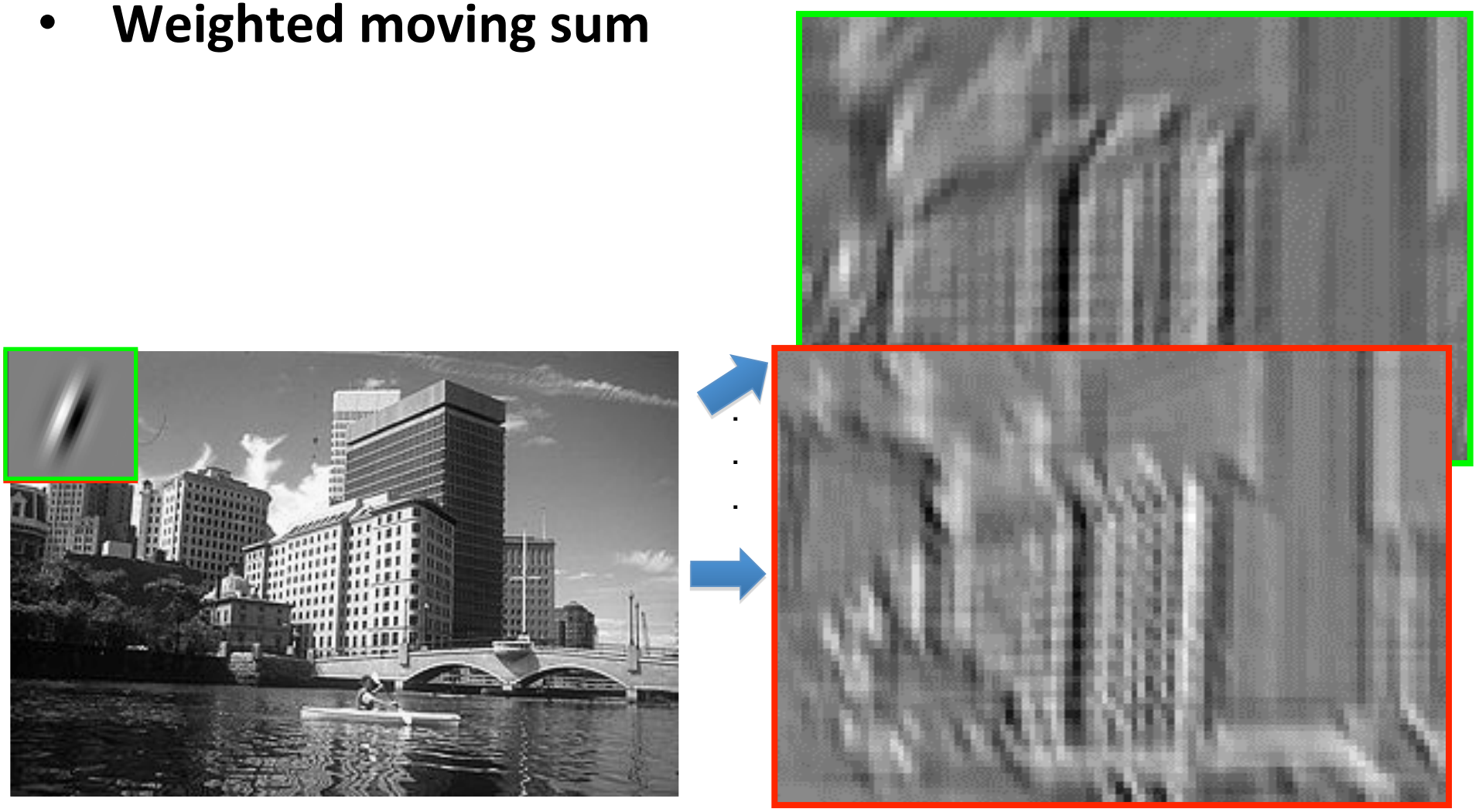
0	0	0
-1	-2	-1

Sobel

Horizontal Edge (absolute value)

What is 2D Convolution?

- **Weighted moving sum**

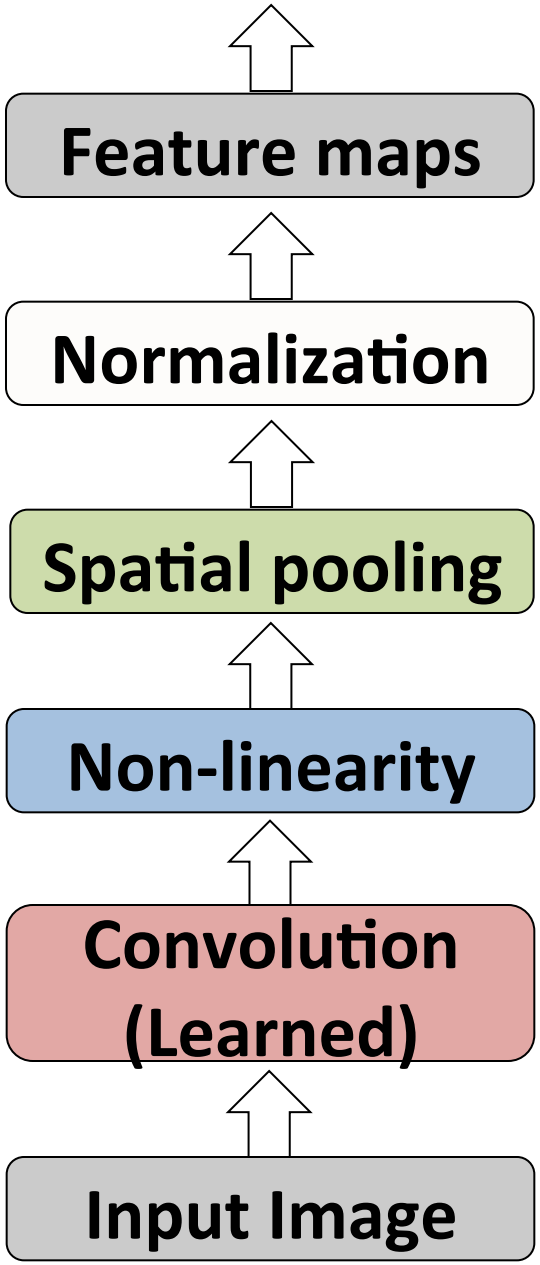


Input

Feature Activation Map

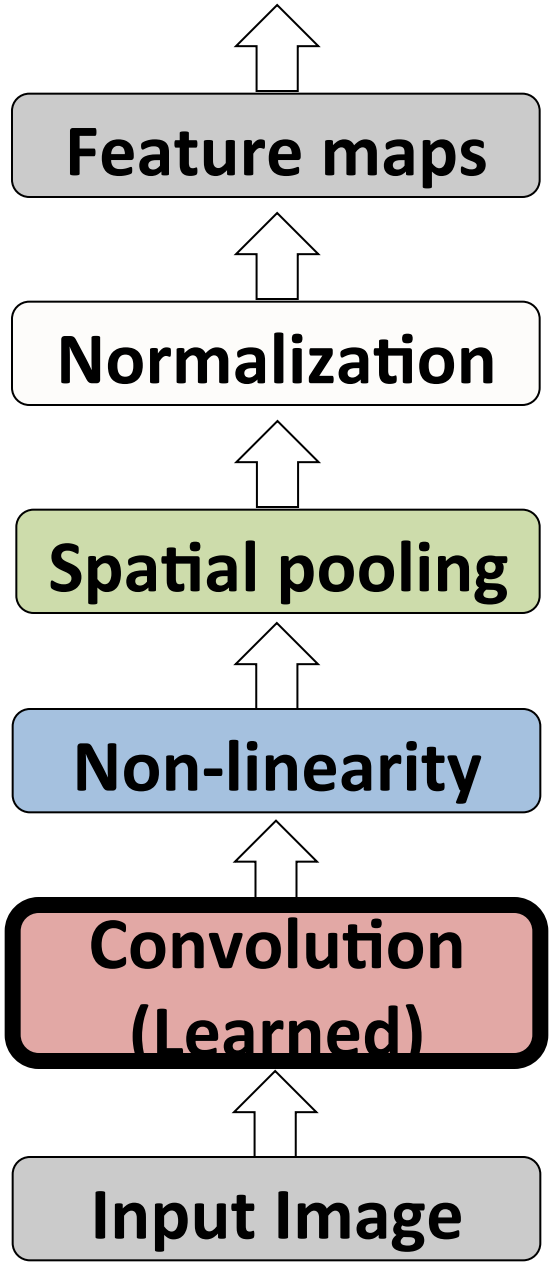
slide credit: S. Lazebnik

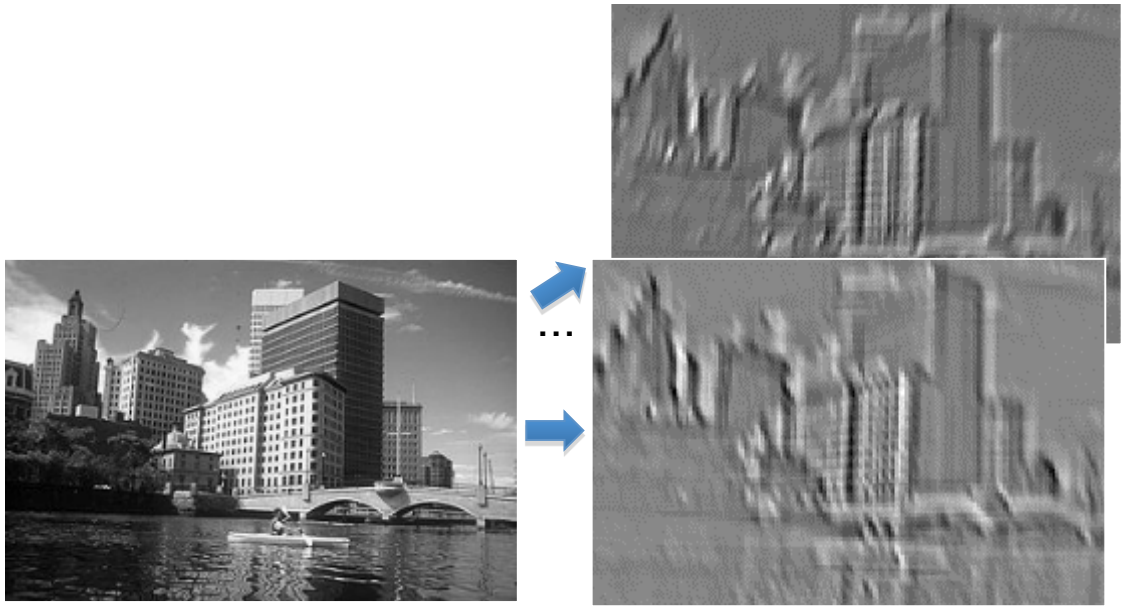
Convolutional Neural Networks



slide credit: S. Lazebnik

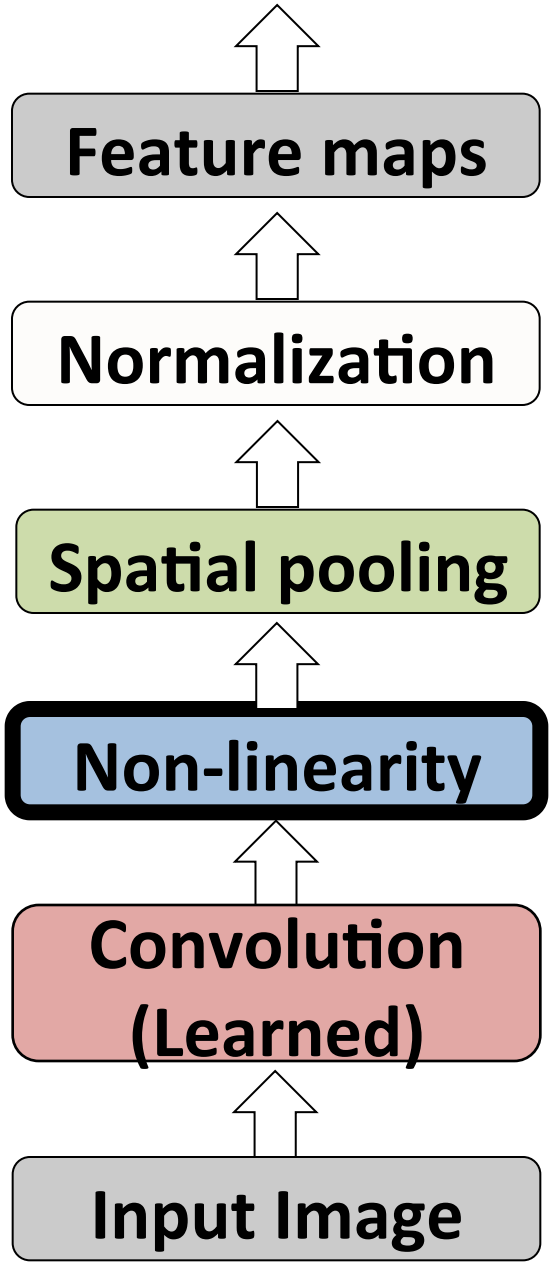
Convolutional Neural Networks

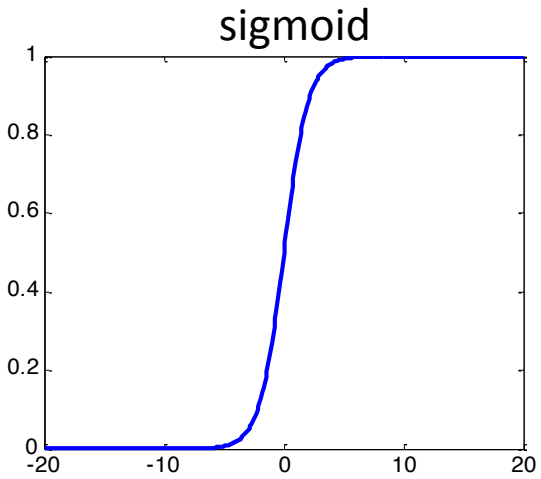


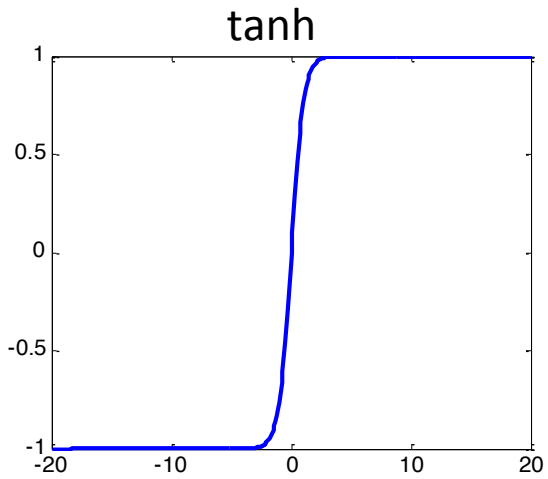


slide credit: S. Lazebnik

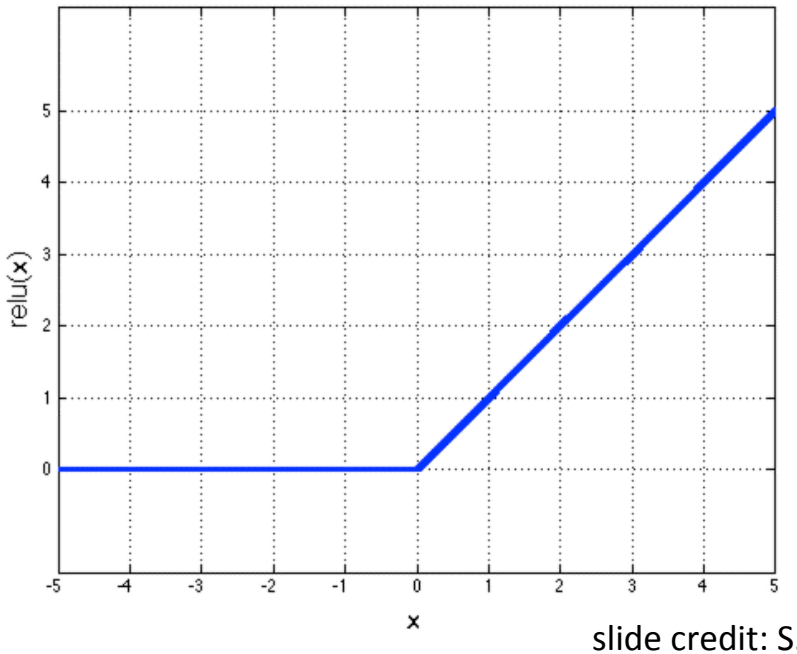
Convolutional Neural Networks





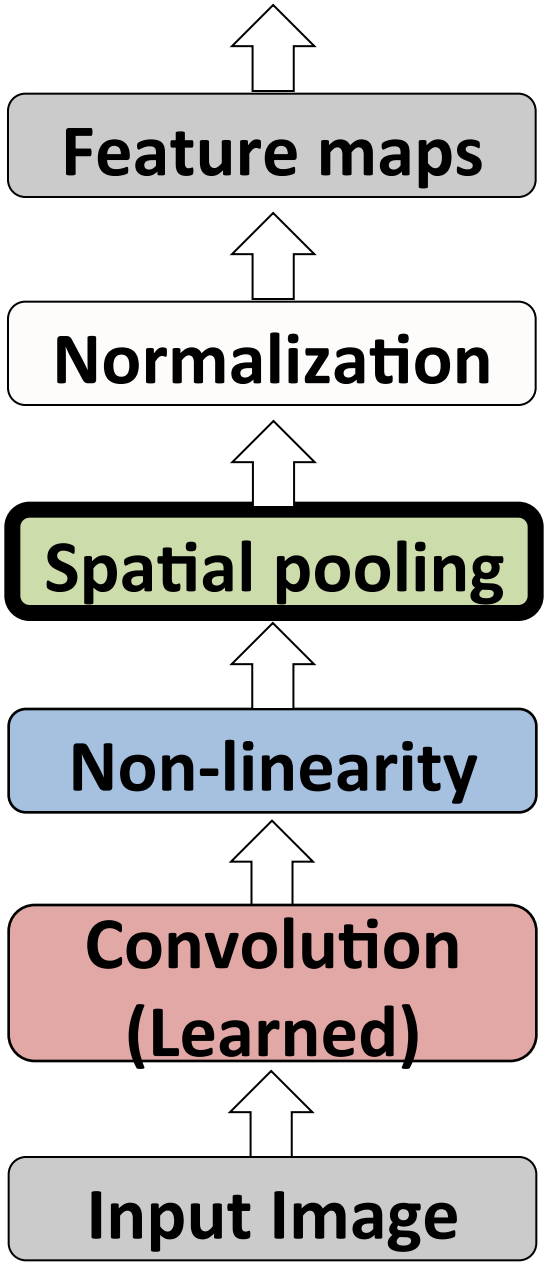


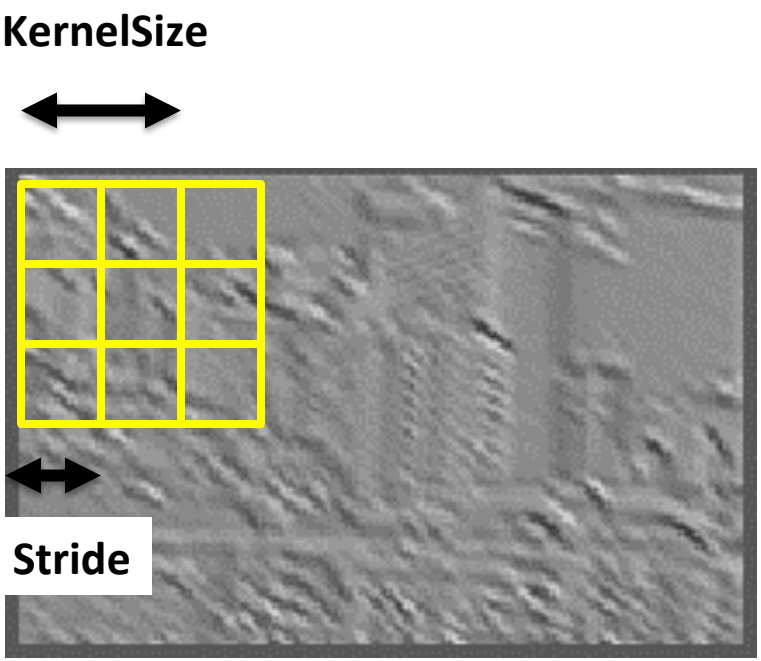
Rectified Linear Unit (ReLU)



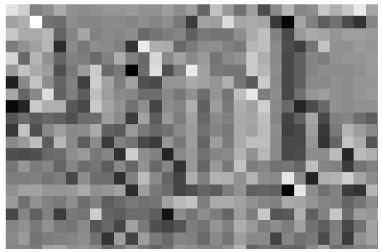
slide credit: S. Lazebnik

Convolutional Neural Networks





Max pooling

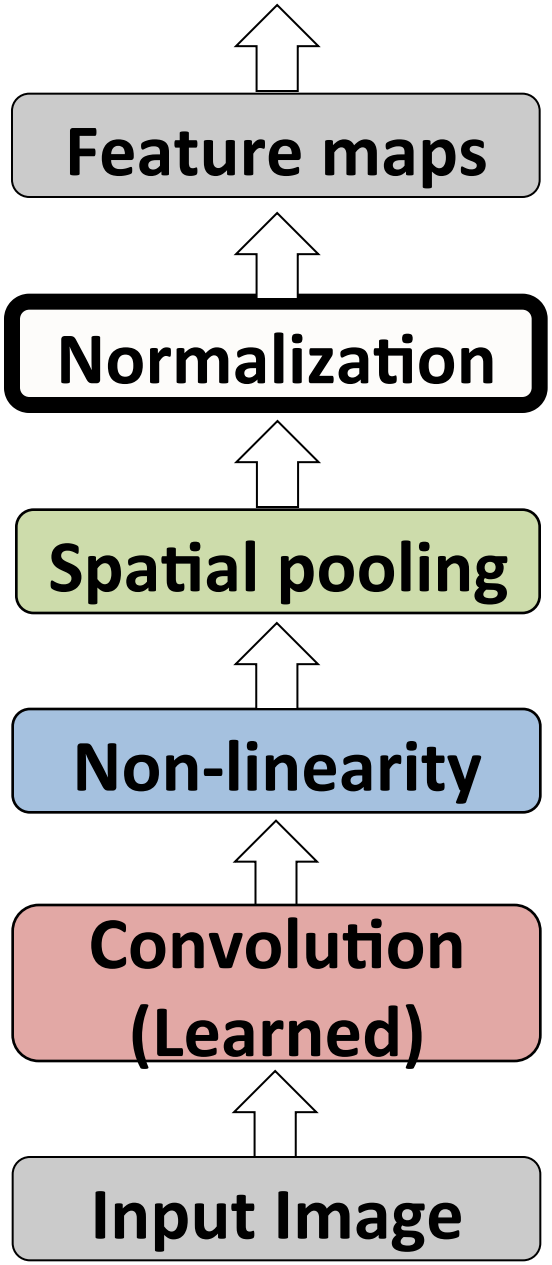


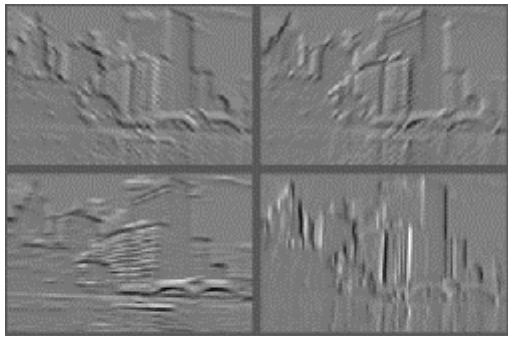
Max-pooling:
a **non-linear** down-sampling

Provide *translation invariance*

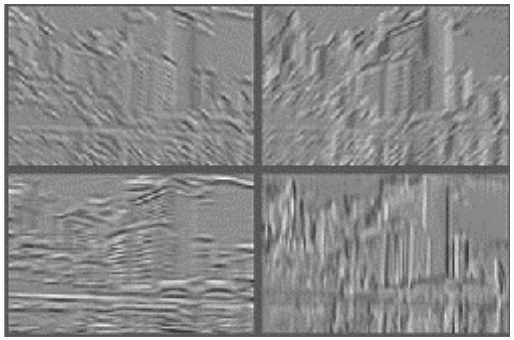
slide credit: S. Lazebnik

Convolutional Neural Networks





Feature Maps



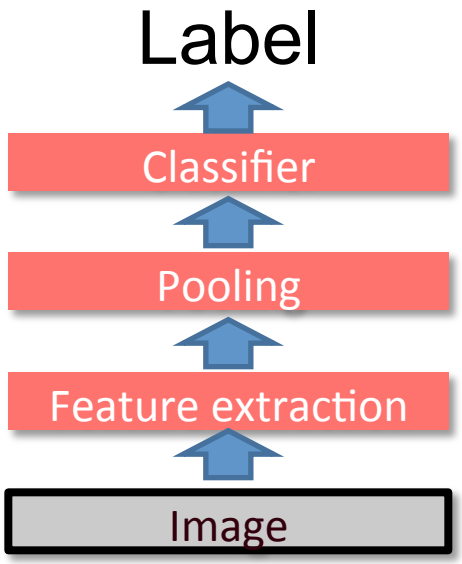
Feature Maps After Contrast Normalization

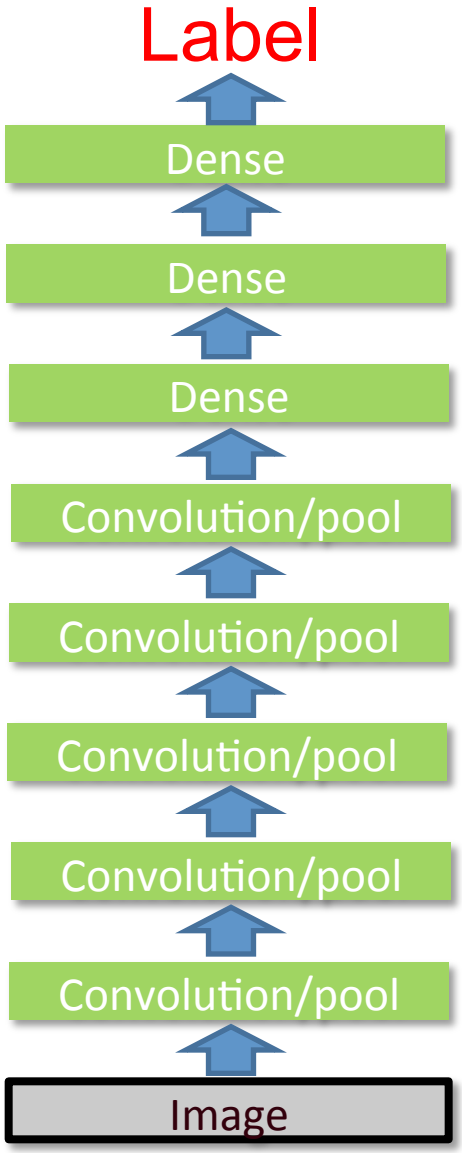
Equalizes the feature maps locally: weak signals are boosted, whereas strong signals are suppressed.

slide credit: S. Lazebnik

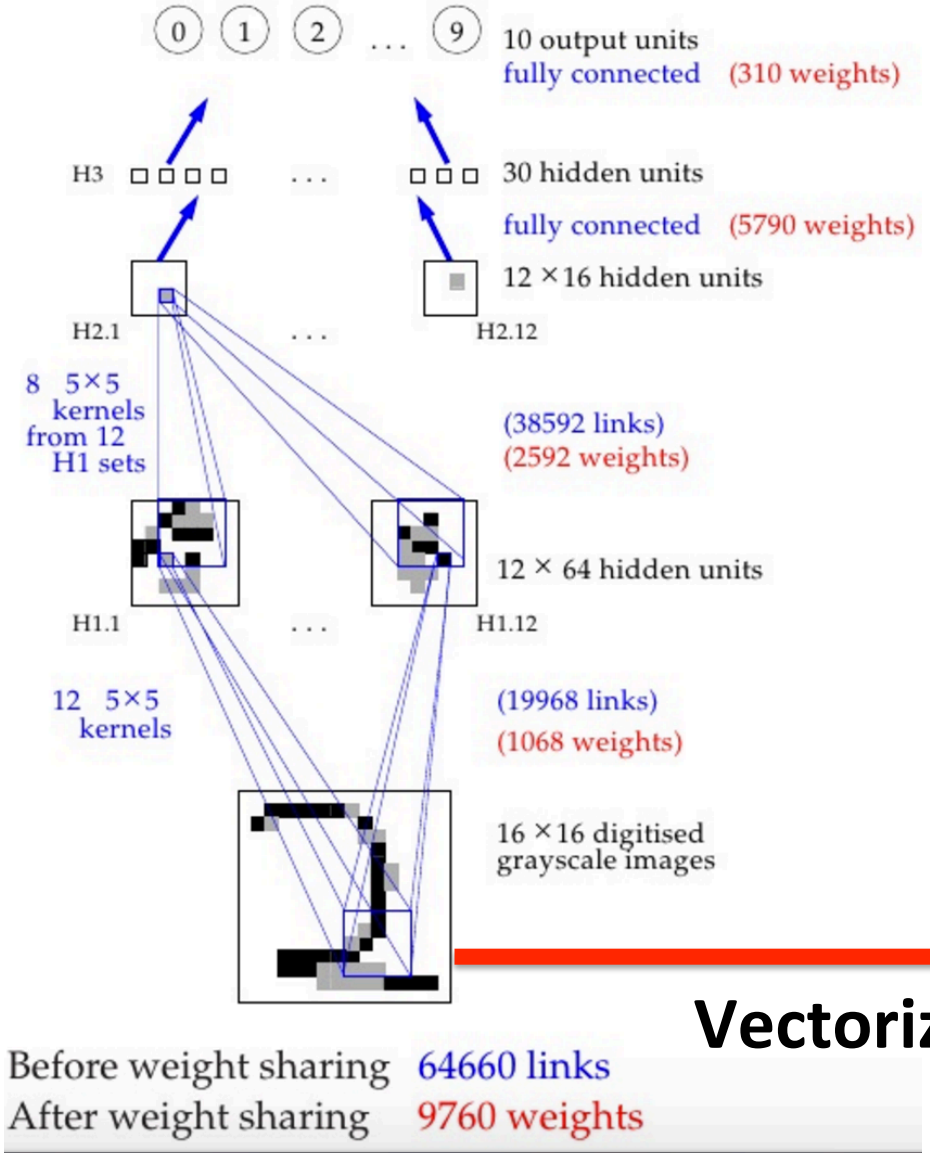
Engineered vs. learned features

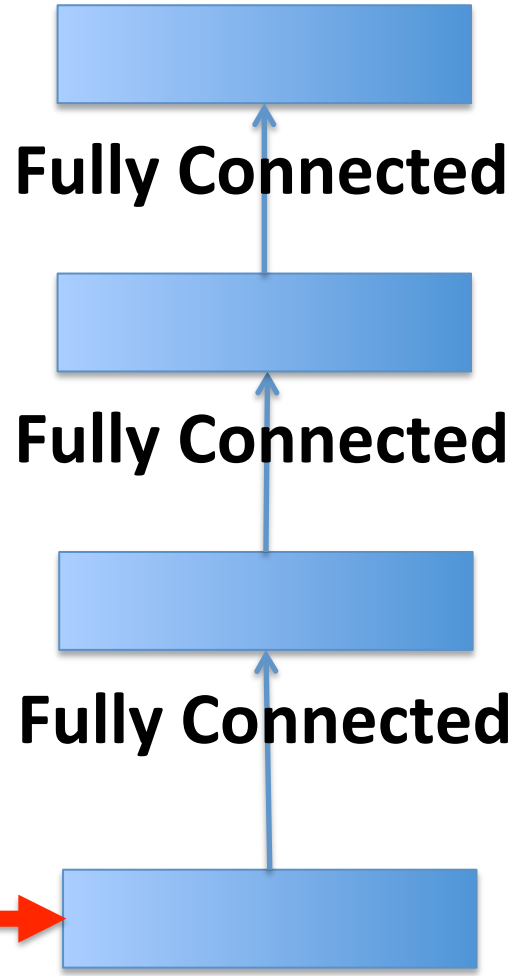
Convolutional filters are trained in a supervised manner by back-propagating classification error





Why Convolution?



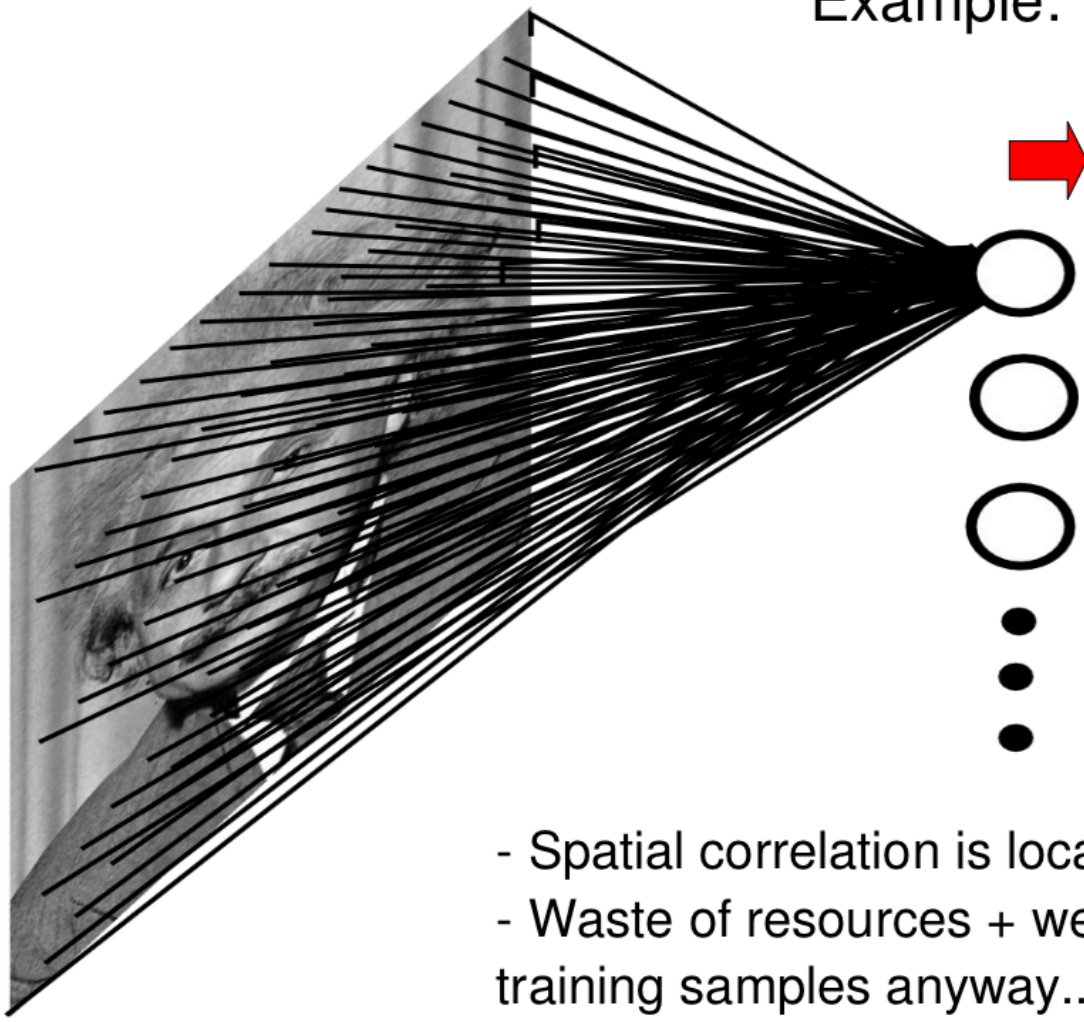


Vectorize

16x16=256

Yann Lecun, 1989

Fully Connected Layer



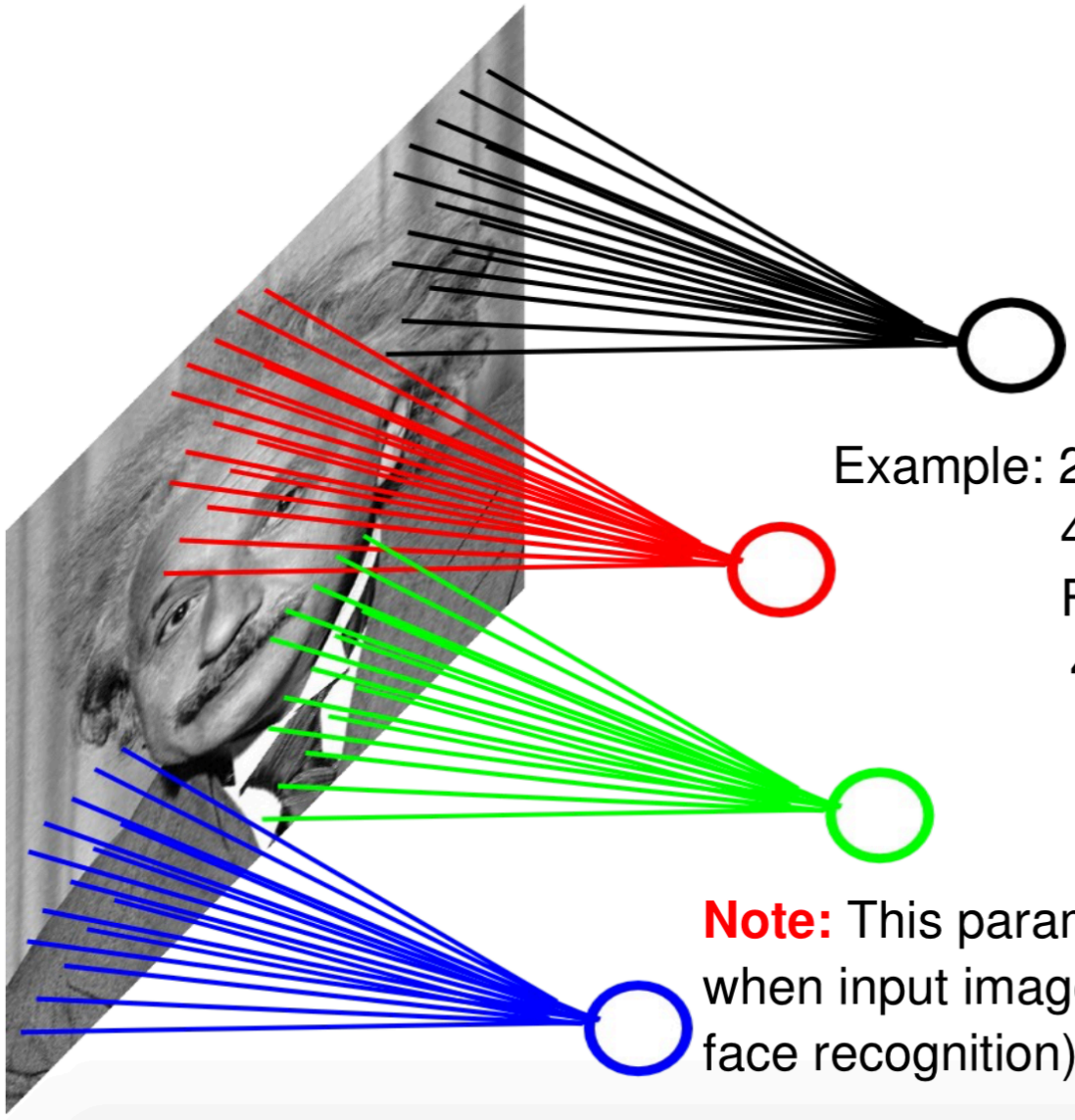
Example: 200x200 image

40K hidden units

~2B parameters!!!

- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

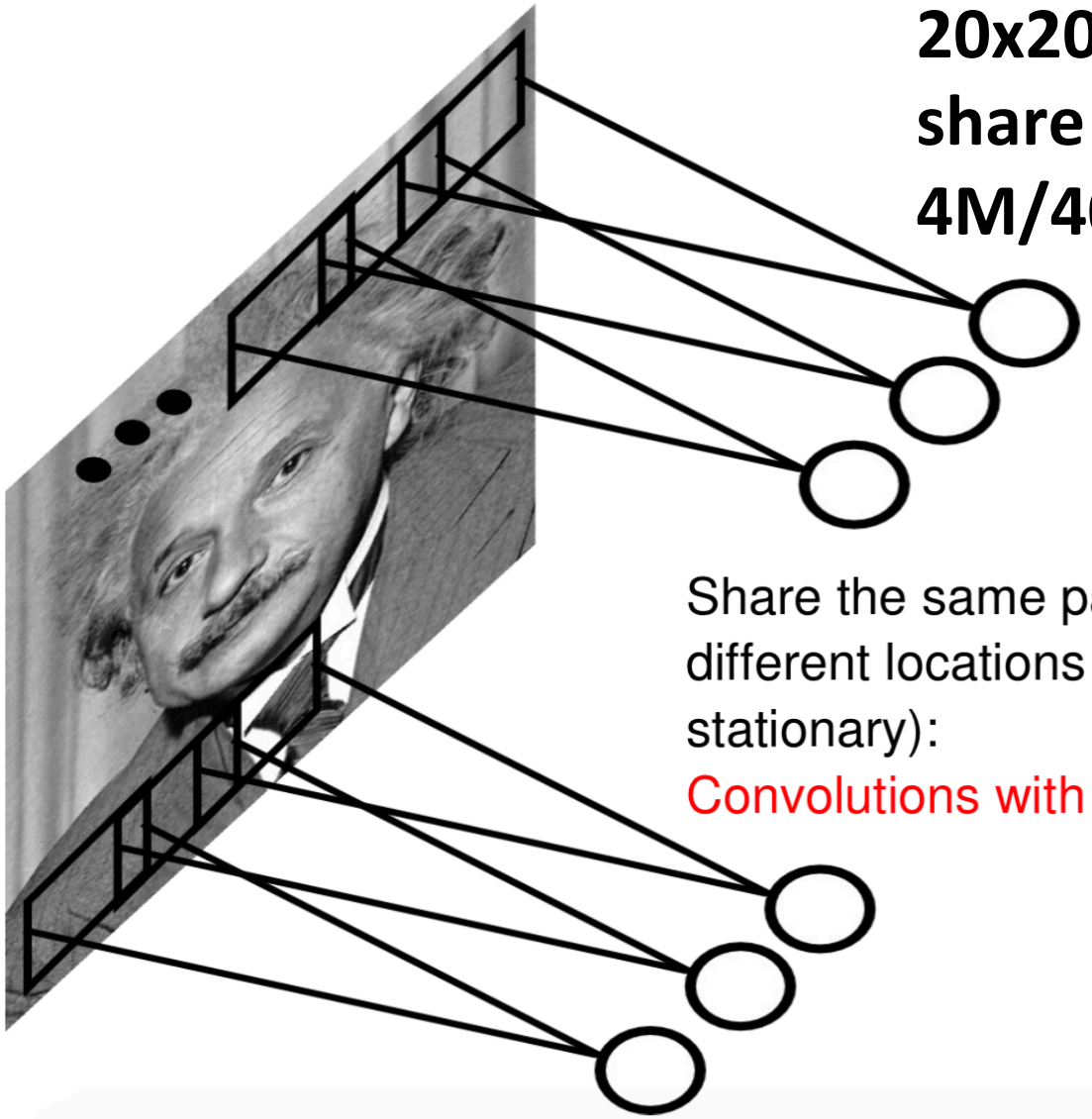
Locally Connected Layer



Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

Convolutional Layer

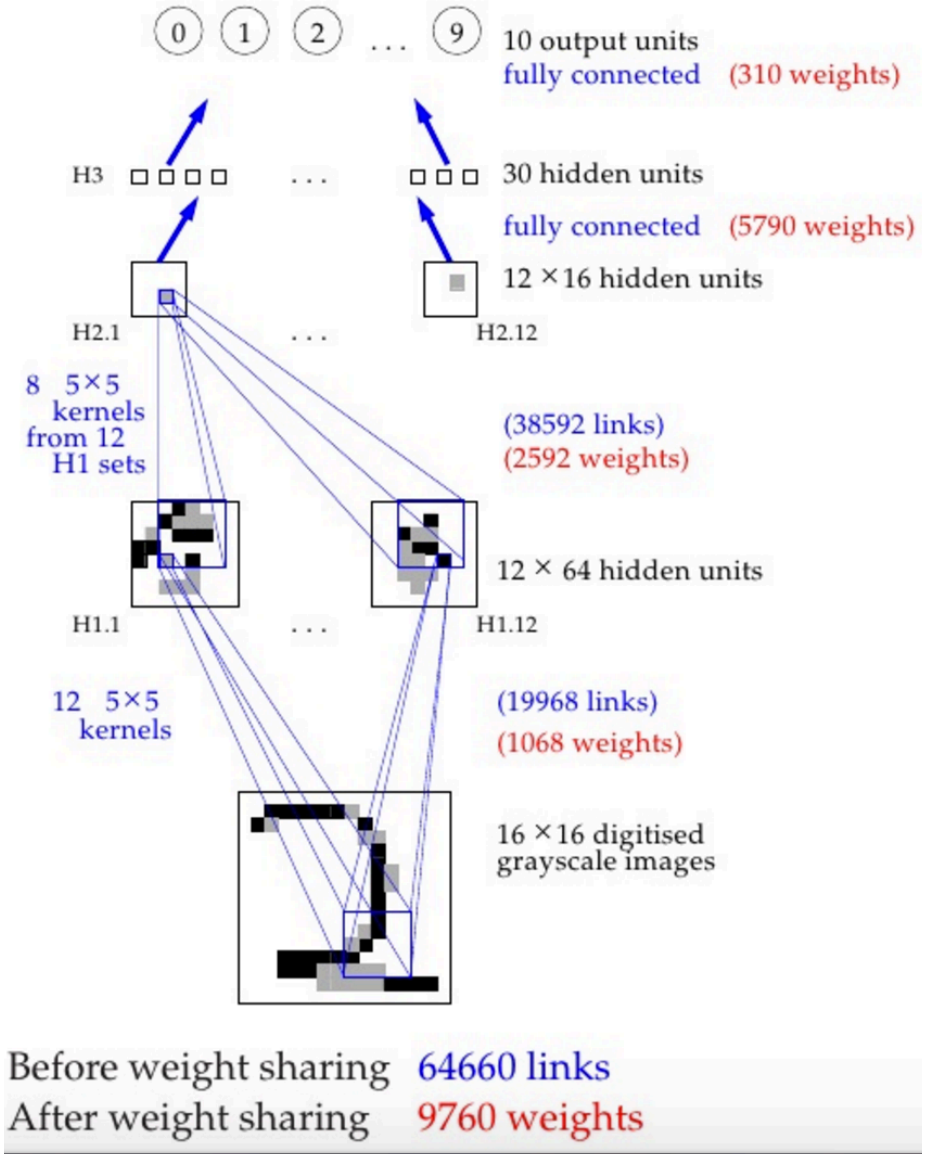


20x20=400 hidden units
share the same parameters
4M/400=10K parameters

2B->4M->10K

Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

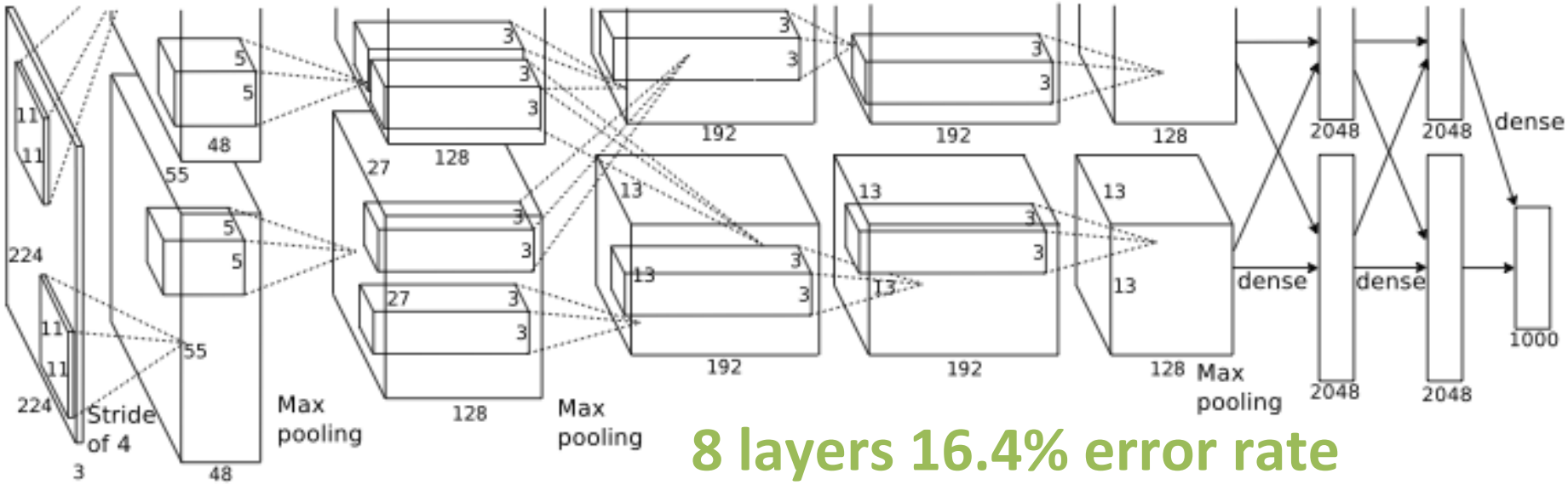
LeNet Details



- **Input**: **256**
- **Weights**: **9760**
- Train on 7K examples
- Train on SUN-4/260
- Test on AT&T DSP-32C

Yann Lecun, 1989

Modern CNN: AlexNet



8 layers 16.4% error rate

Input: 224*224*3=**150K (v.s. 256 LeNet)**

Neurons: 290400+186624+64896+64896+43264+4096+4096+1000=**650K**

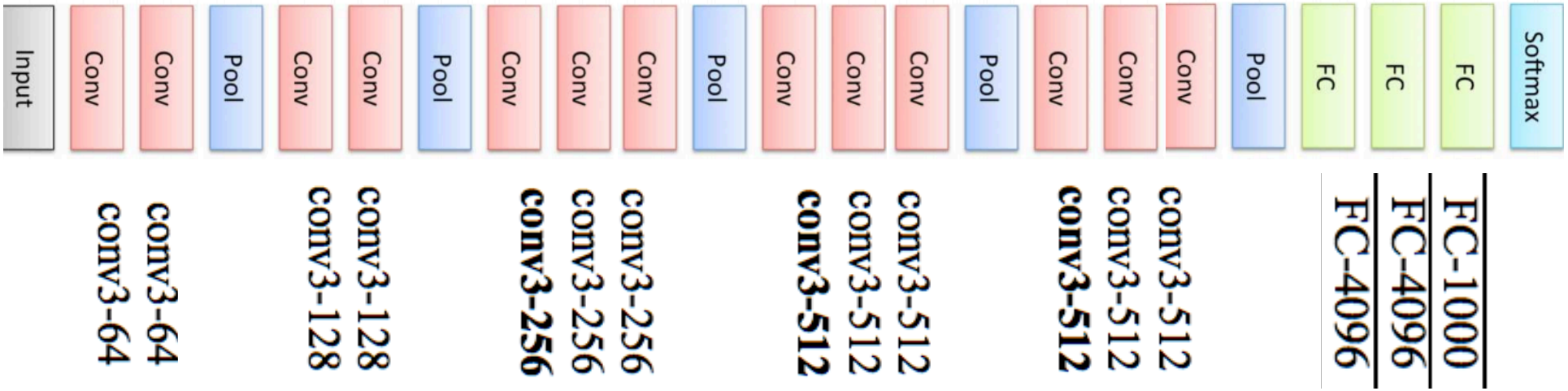
Weights: 11*11*3*48*2(35K)+5*5*48*128*2(307K)+128*3*3*192*4(884K)+ 192*3*3*192*2(663K)+192*3*3*128*2(442K)+6*6*128*2048*4(38M) +4096*4096(17M)+4096*1000(4M)=**60M (v.s. 9760 LeNet)**

- **More data (1.2M vs. 7K images)**
- **Trained on two GPUs for a week**
- **Dropout**

Krizhevsky et al. "ImageNet classification with deep convolutional neural networks," NIPS, 2012.

Modified from Tsung-Han Chan's slides

Modern CNN: VGG-16



"conv<kernel size>-<number of channels>"

8 layers ➡ 16 layers

16.4% error rate 7.3% error rate

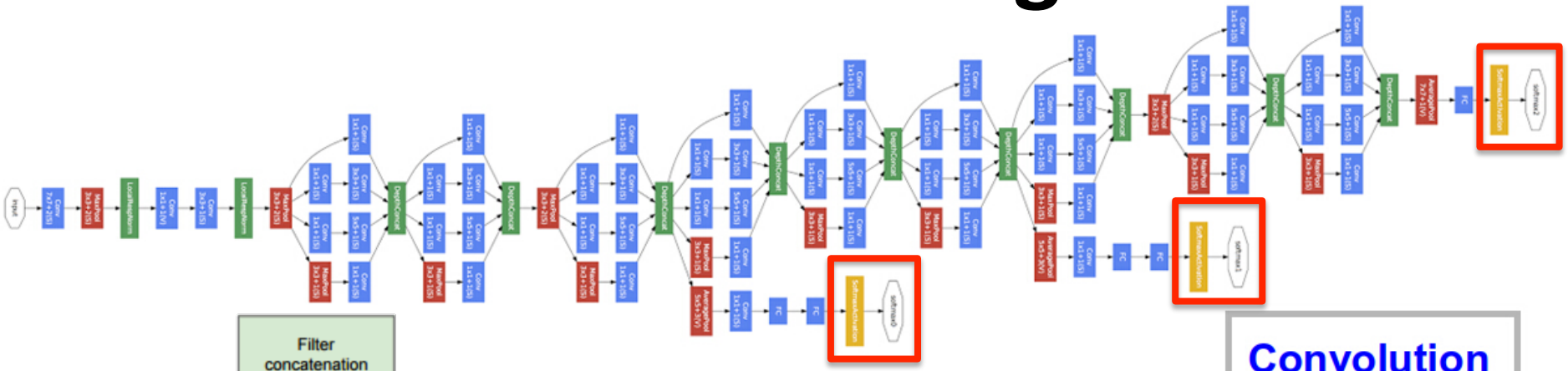
Input: 224*224*3=**150K**

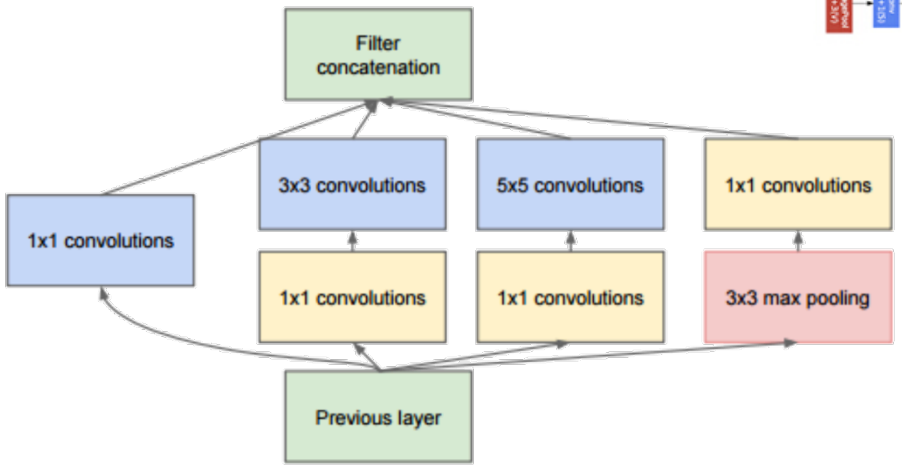
Weights: [2K+37K]+[74K+148K]+[295K+*590K*+590K]+[1M+2M+2M]+[2M+2M+2M]+[103M+17M+4M]=**138M**

- **Small kernel size**
- **Large #channels (512)**
- **No Local Response Normalization (LRN)**

Simonyan and Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," arxiv :1409.1556v6, Sept. 2014

Modern CNN: GoogLeNet





Convolution
Pooling
Softmax
Other

16 layers ➡ 22 layers

7.3% error rate 6.66% error rate

Inception module with dimension reductions

Input: 224*224*3=**150K**

Weights: ***6,797,700=6M (v.s. 138M VGG-16 and 60M AlexNet)***

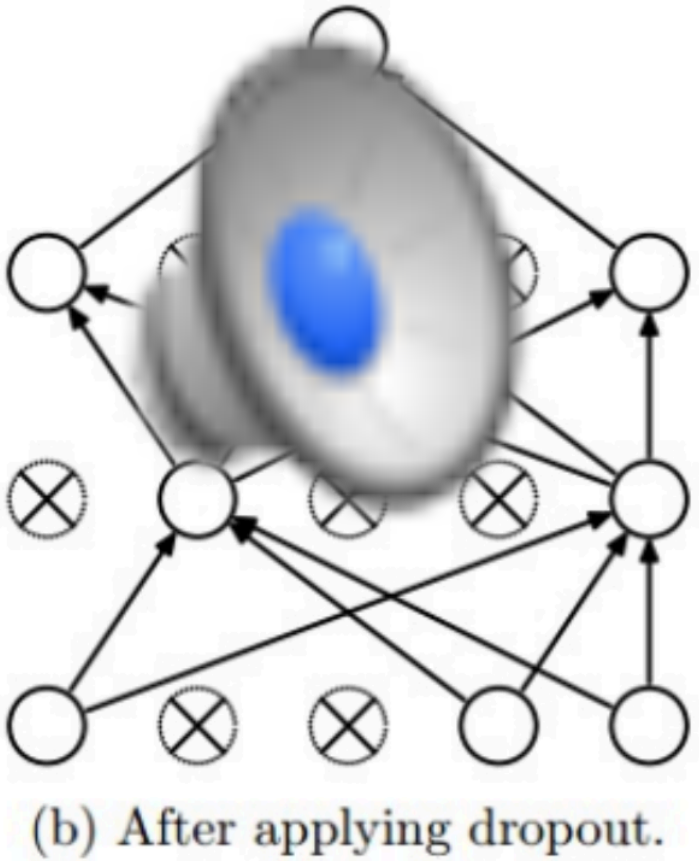
- **A computational budget of 1.5 billion multiply-add**
- **Multiple layer of outputs**

Christian Szegedy et al., "Going deeper with convolutions," arXiv:1409.4842v1, Sept. 2014.

Do Better: Random in Forward-Pass

- **Dropout**

(a) Standard Neural Net

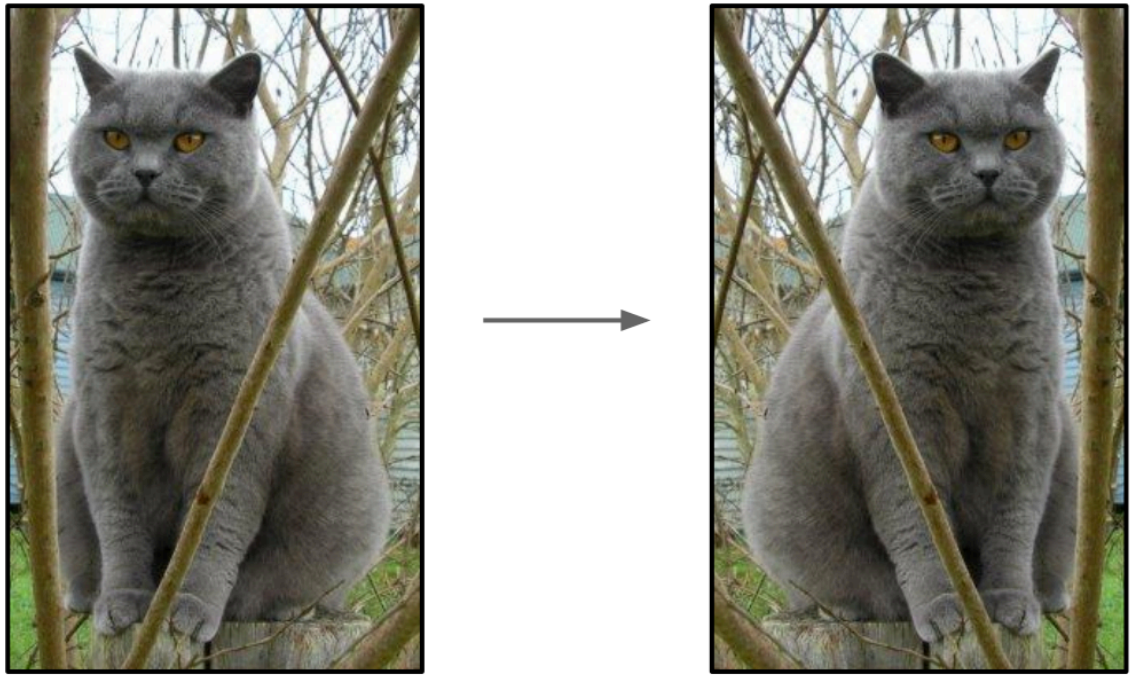


(b) After applying dropout.

Krizhevsky et al. “ImageNet classification with deep convolutional neural networks,” NIPS, 2012.

Do Better: Data Augmentation

- Simulating “fake” data
- Explicitly encoding image transformations that shouldn’t change object identity.
  - **Flip horizontally**

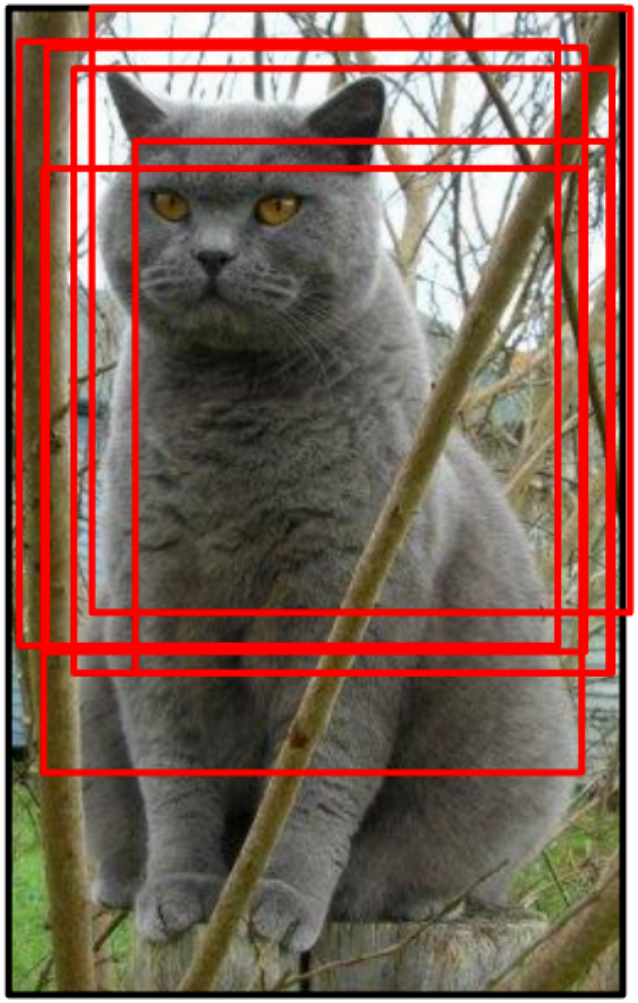


Slides from
Fei-Fei and Karpathy

Do Better: Data Augmentation

– **Random/Multiple crops/ scales**

Common up to 150 or more crops



Slides from
Fei-Fei and Karpathy

Do Better: Data Augmentation

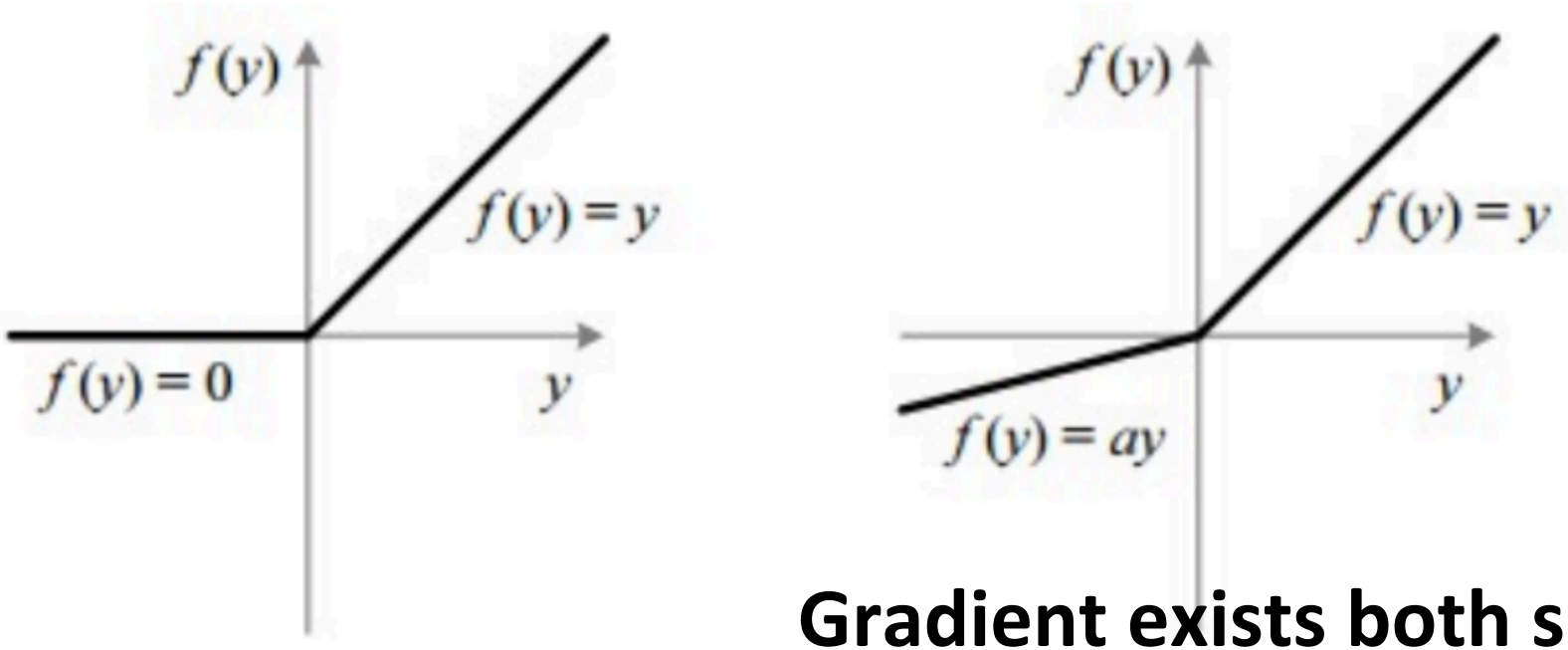
- Random mix/combinations of :
  - Translation
  - Rotation
  - Stretching
  - Shearing,
  - Lens distortions
  - Color jittering

- DeepImage – Baidu 2015
  - Data augmentation + multi GPU (error **5.3**%)

Slides from
Fei-Fei and Karpathy

Ren We et al., "Deep Image: Scaling up Image Recognition," arXiv:1501.02876v1, Jan. 2015.

Do Better: Parametric ReLU



Gradient exists both side

\+ Careful initialization of the weights

- **PReLU-nets - MSRA (error 4.9%)**

Slides from
Fei-Fei and Karpathy

Kaiming He et al., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," arXiv:1502.01852v1, Feb. 2015.

Do Better: Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

- Allow higher learning rates
- Less careful initialization
- some cases eliminating the need for Dropout
- error **4.8**%

Modified from
Jia-Bin Huang

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Ioffe and Szegedy 2015]

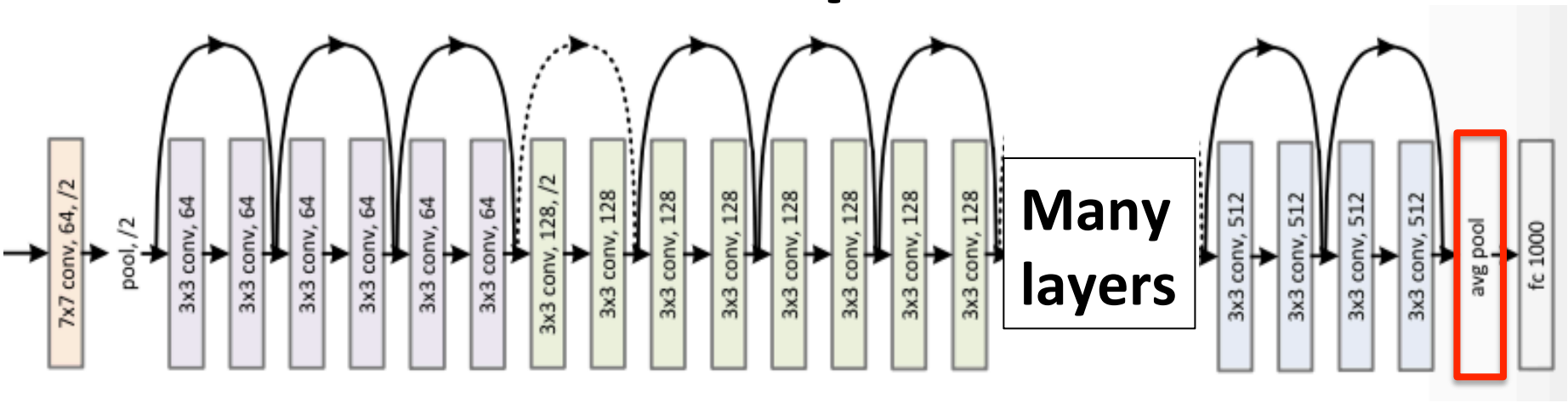
ILSVRC 2015 Detection

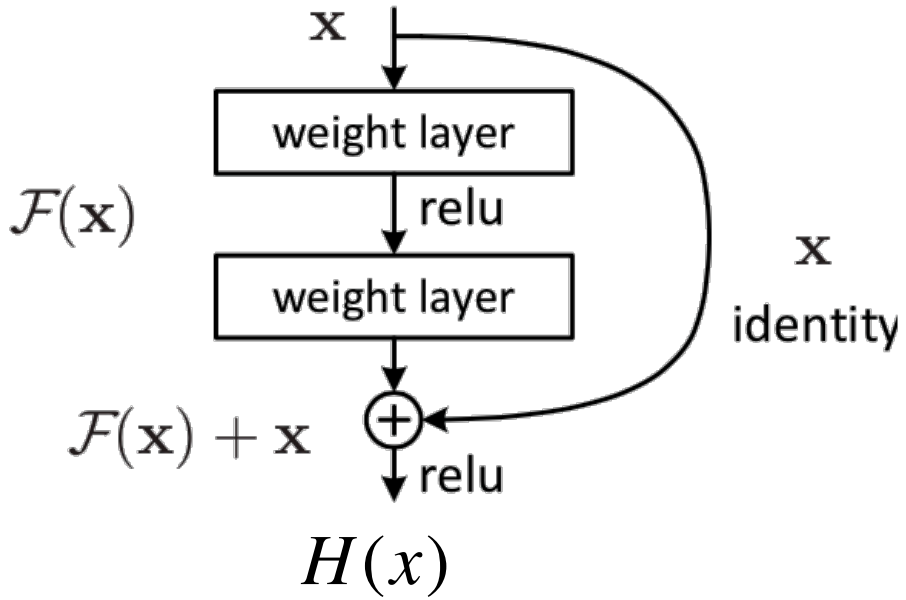
Ordered by mean average precision

Team name	Entry description	**mean AP**	Number of object categories won
MSRA	An ensemble for detection.	0.620741	194
MSRA	*A single model for detection.*	*0.588451*	---
Qualcomm Research	NeoNet ensemble with bounding box regression. Validation mAP is 54.6	0.535745	4
Qualcomm Research	*NeoNet ensemble without bounding box regression. Validation mAP is 53.6*	*0.531957*	---
CUimage	Combined multiple models with the region proposals of cascaded RPN, 57.3% mAP on Val2.	0.527113	2
The University of Adelaide	9 models	0.514434	0

Top-5 **3.57**% error

Winner 2015: Deep Residual Nets





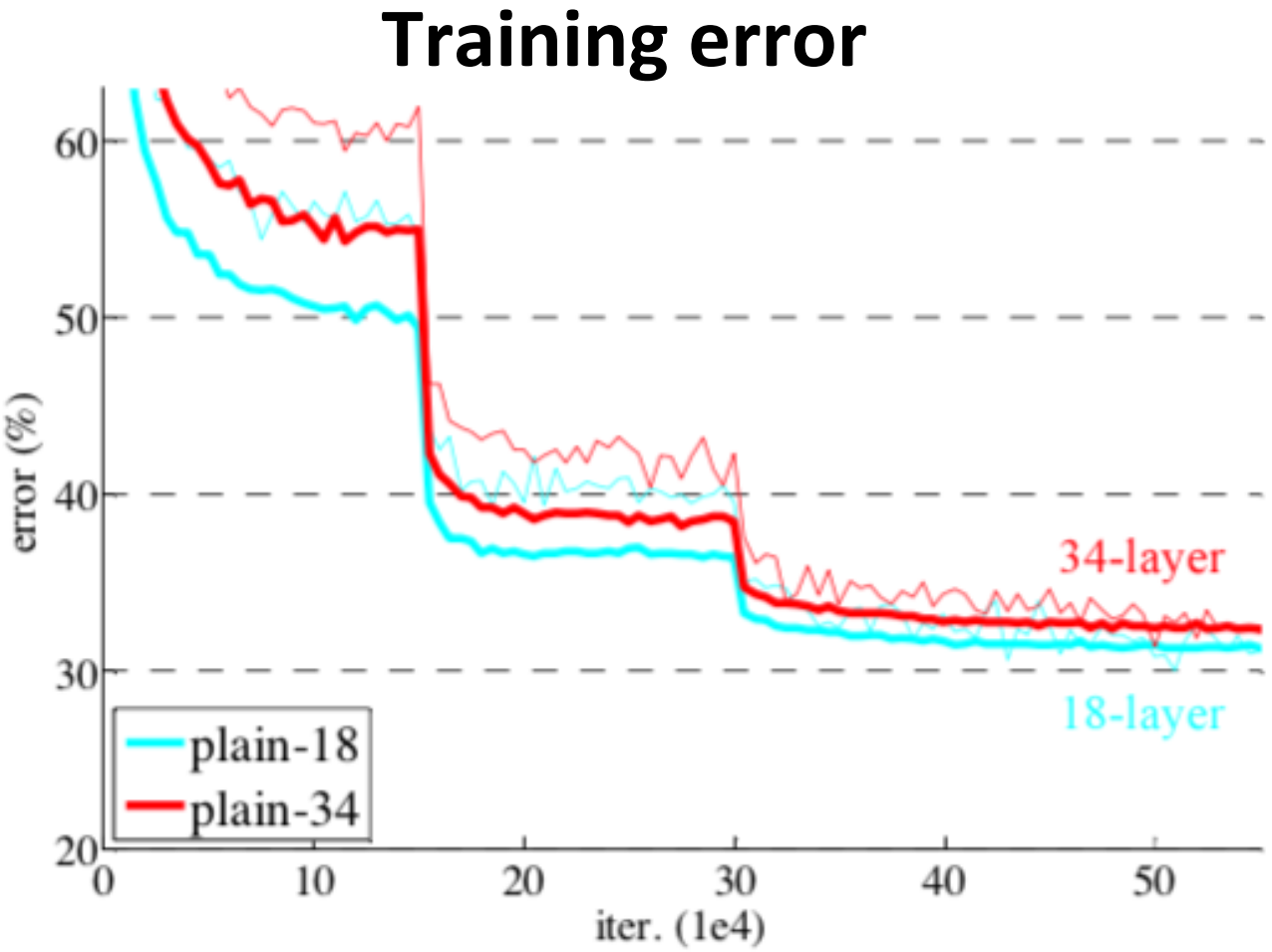
$$H(x) = F(x) + x$$

$$F(x) = H(x) - x$$

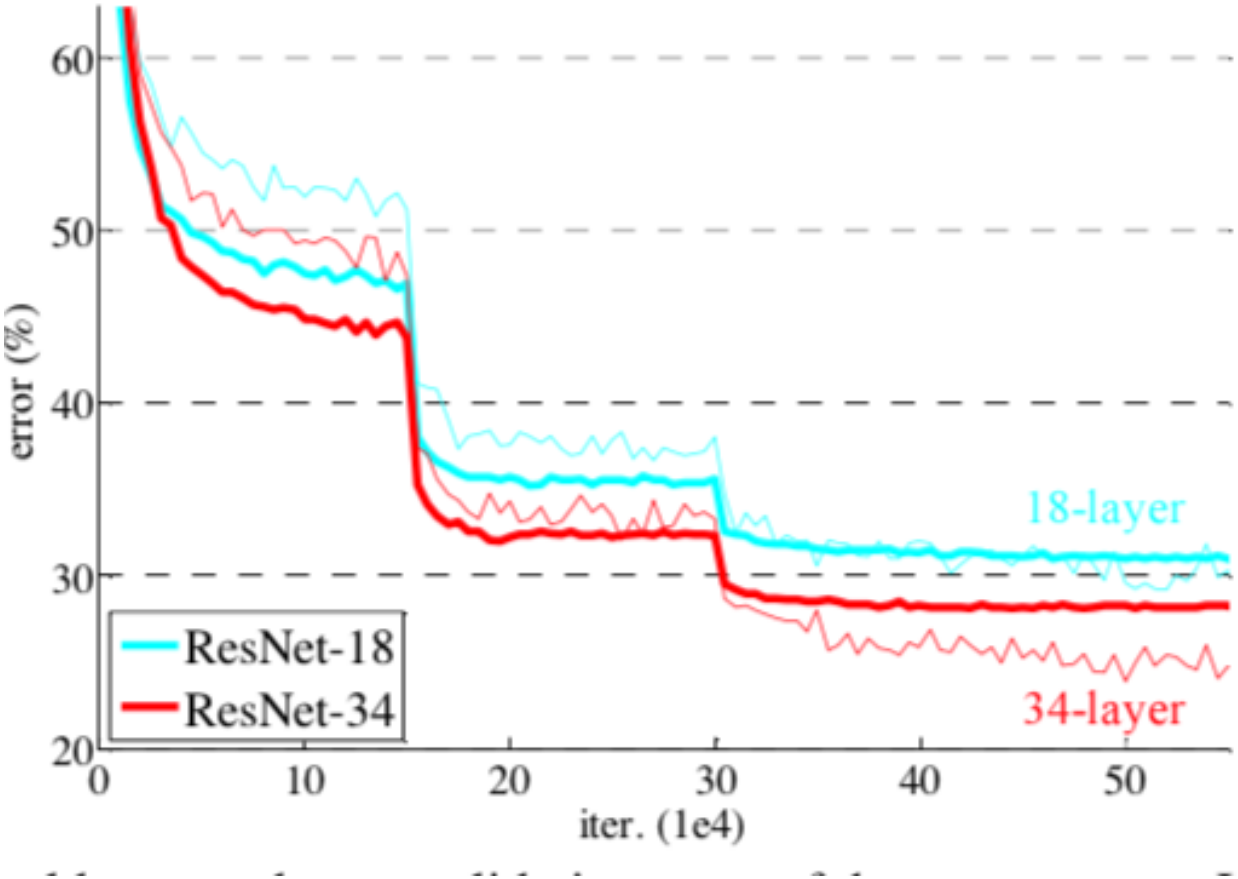
- 152 layers- 8x deeper than VGG-19

Kaiming He et al., "Deep Residual Learning for Image Recognition," arXiv:1512.03385v1, Dec. 2015.

With Fix Data, Deeper the Better?



Training error



Beyond ImageNet Classification

- Detection
- Segmentation
- Regression
- Pose estimation
- Matching patches
- ……Many More……

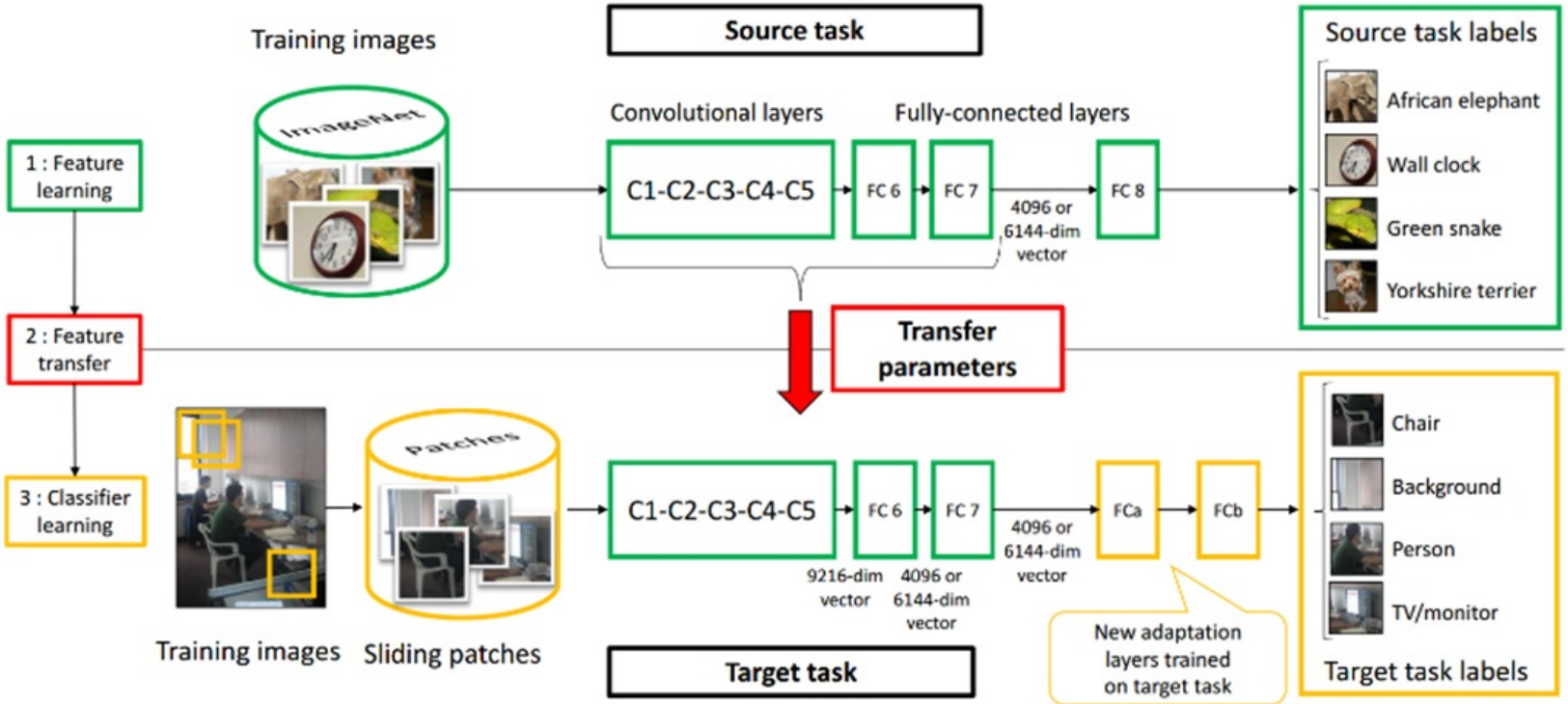
Key Enablers

Do we need a lot of data to train a CNN?

Not really!

Transfer Learning

- Improvement of learning in a **new** task through the *transfer of knowledge* from a **related** task that has already been learned.
- Weight initialization for CNN



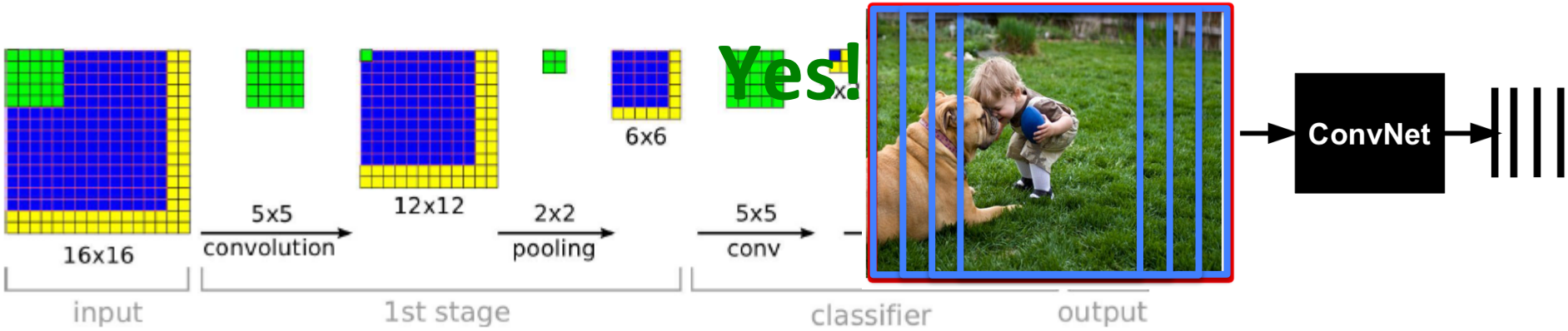
Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks [Oquab et al. CVPR 2014]

Slides from Jia-Bin Huang

- Unrolls convolutions over bigger images
- Produces outputs at several locations with little extra operations

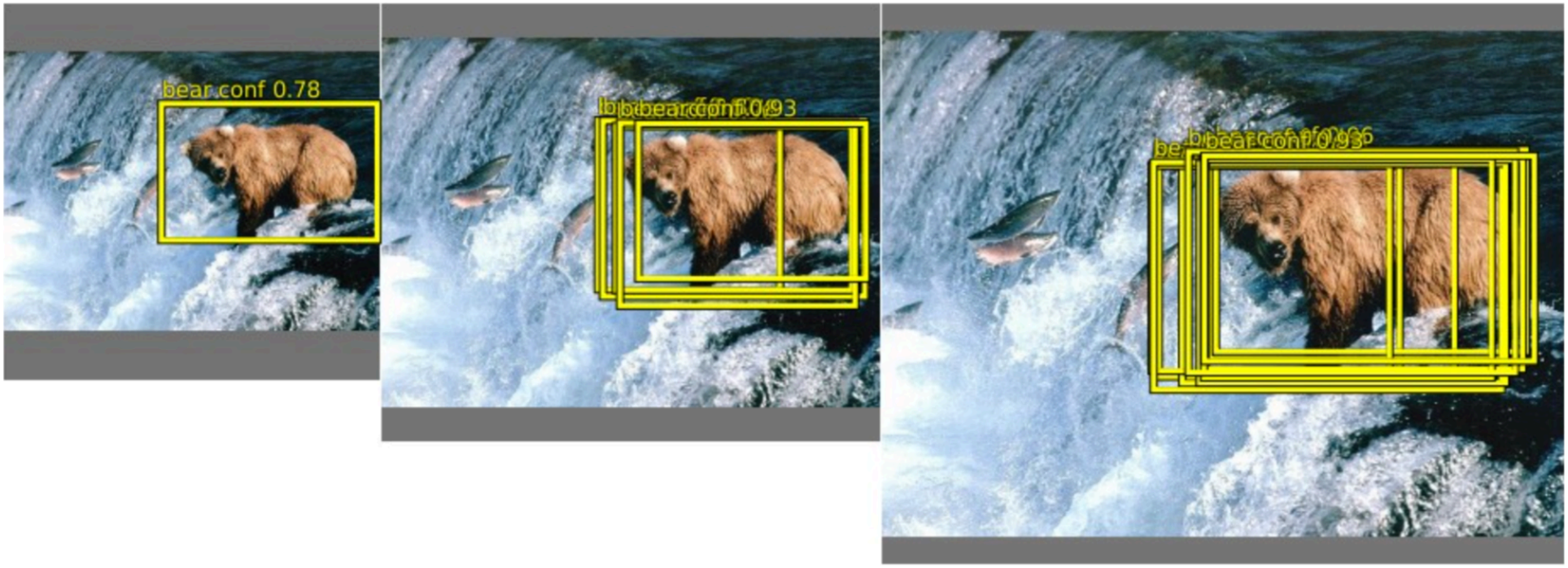
Can CNN handle arbitrary Image size?

Yes!



Object Detection: Overfeat

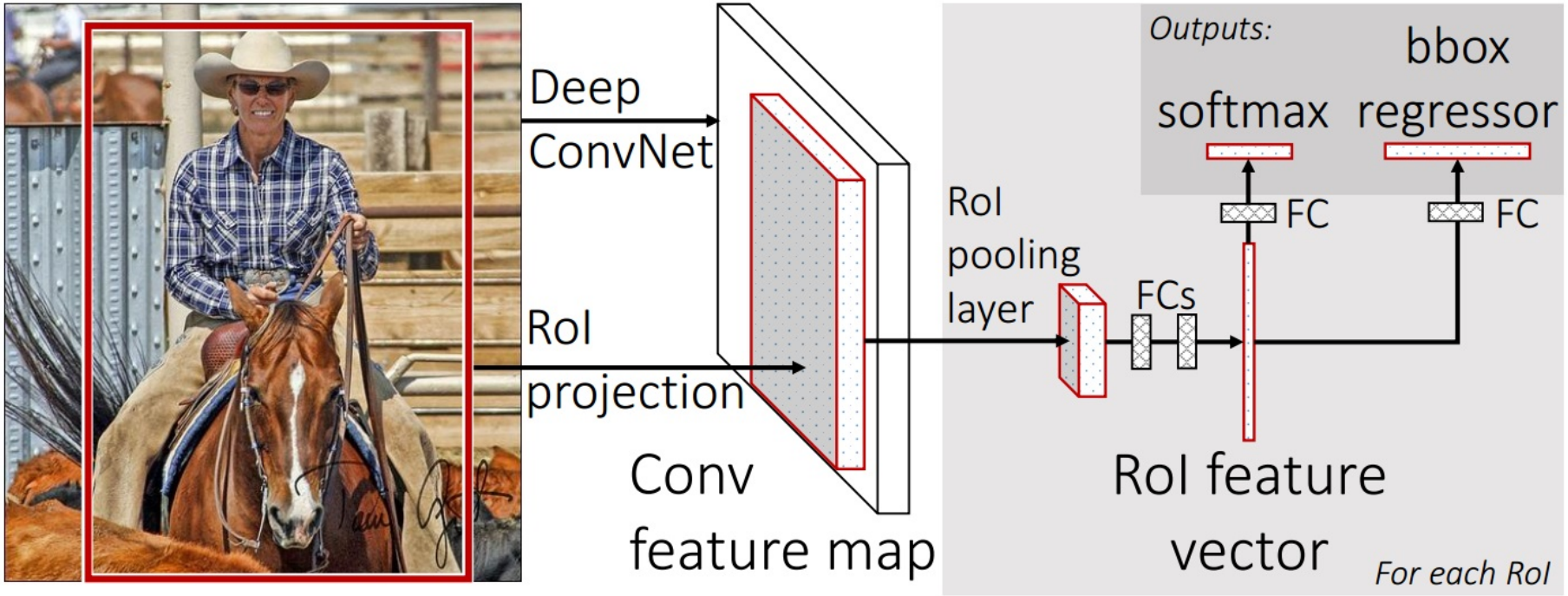
- Objects at multiple scale
  - **One** CovnNet model
  - **Multiple** CovnNet inference
  - **Regress** window location and aspect ratio



Sermanet et al. “OverFeat: Integrated recognition, localization. Arxiv 2013

Object Detection: Fast/Faster R-CNN

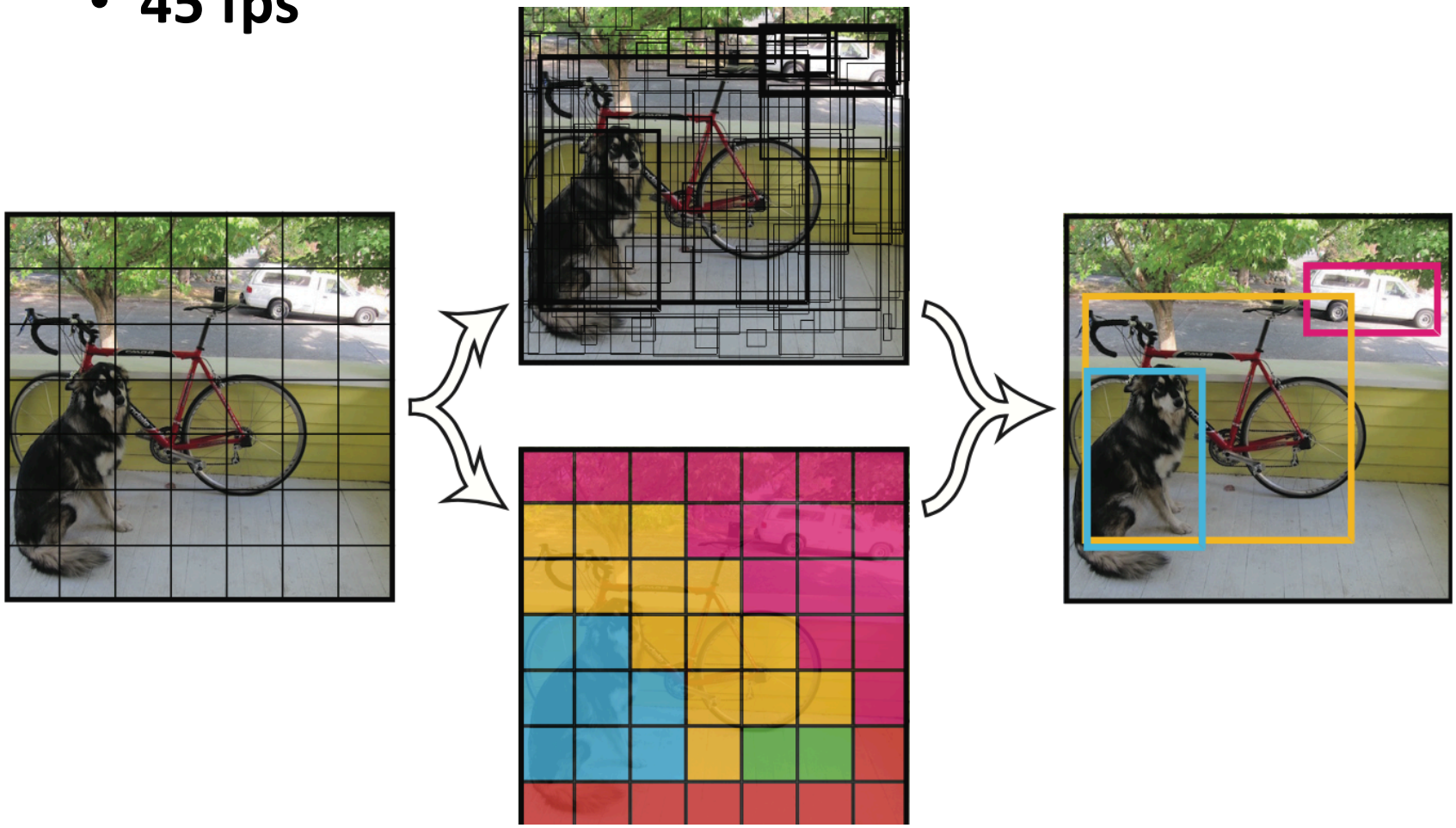
- CovnNet inference at **one** scale
- Region of Interest (**RoI**) pooling with
  - Any size and Any aspect ratio
- 3rd Party region proposal/Region Proposal Networks
- 0.5/5 frame per second (fps)



Fast/Faster RCNN [Girshick et al. 2015, Ren et al. 2015]/
https://github.com/rbgirshick/fast-rcnn https://github.com/ShaoqingRen/faster_rcnn

Object Detection: YOLO

- **Detection as segmentation and regression**
- **45 fps**



Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. Arxiv June 2015

Object Detection: YOLO

- **Detection as segmentation and regression**
- **45 fps**

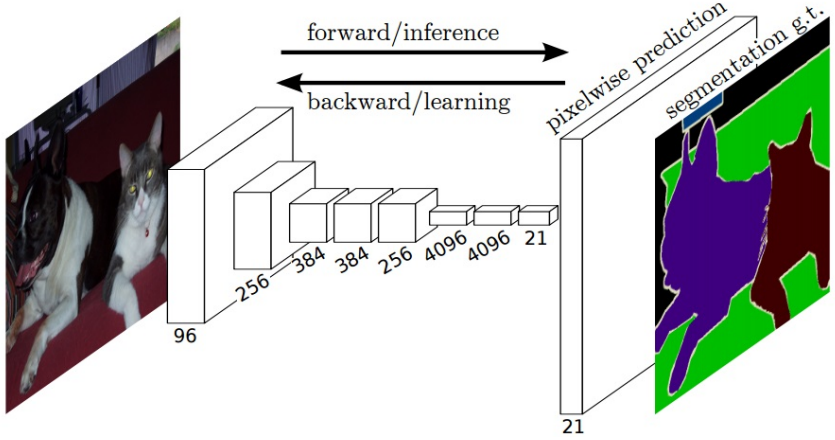
Can CNN generate per pixel output?

Yes!

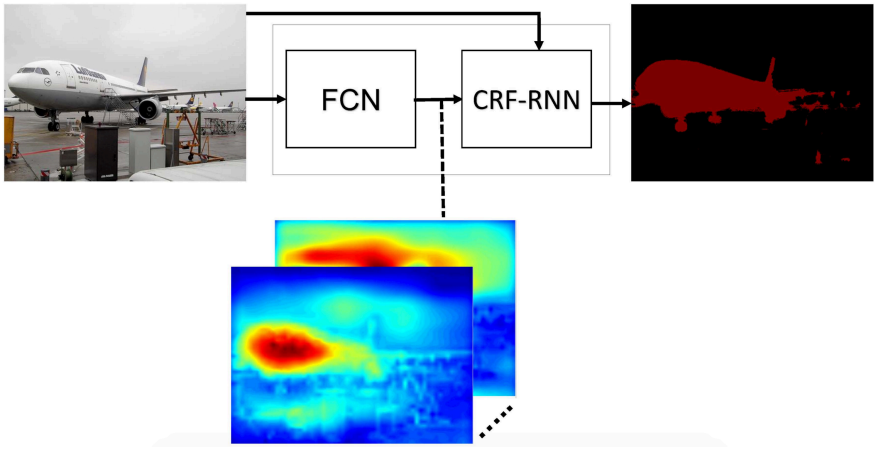
Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. Arxiv June 2015

Dense Prediction – Per Pixel

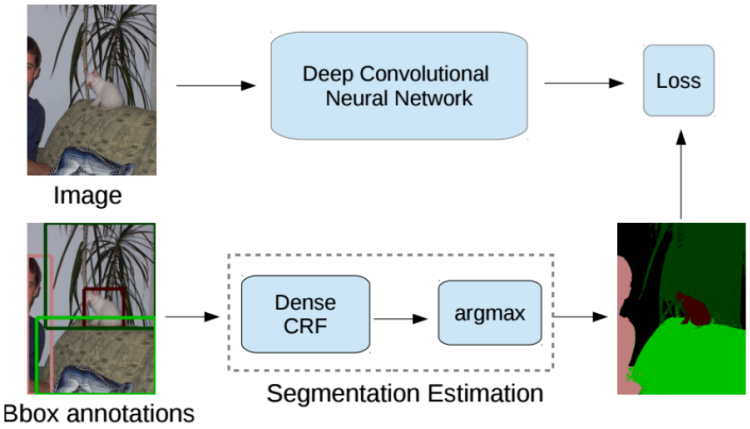
- **Semantic Labeling/Segmentation**



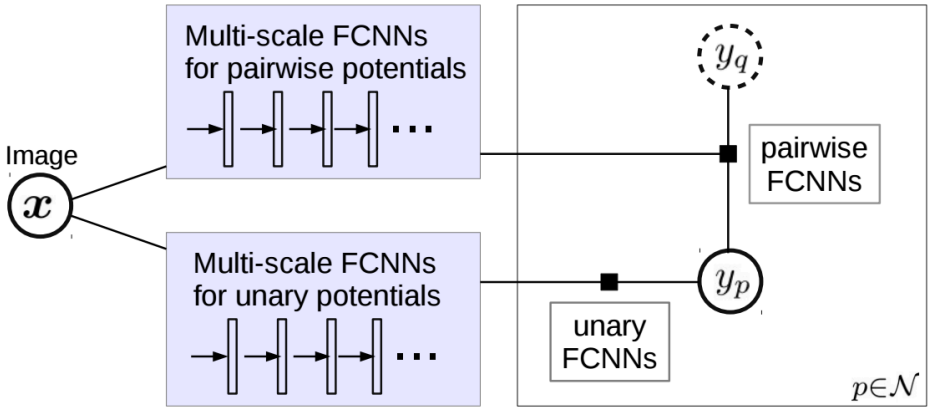
Fully Convolutional Networks for Semantic Segmentation
Long et al. CVPR 2015



Conditional Random Fields as Recurrent Neural Networks
Zheng et al. ICCV 2015



Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation. Papandreou et al. arxiv 2015

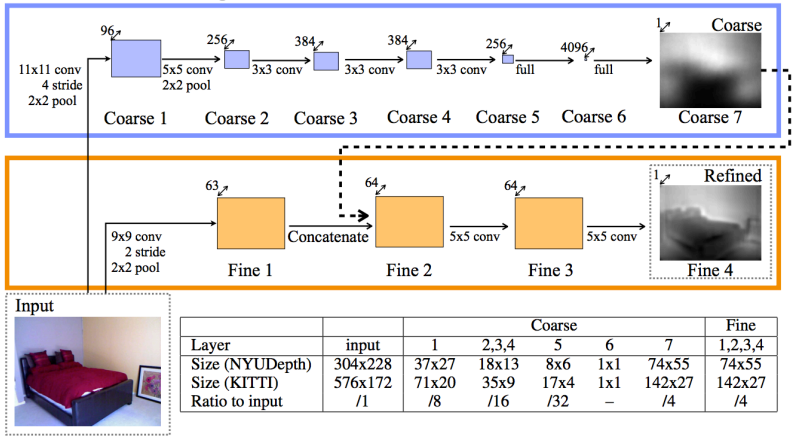


Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation. Lin et al. arxiv 2015

PASCAL Seg - http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=6

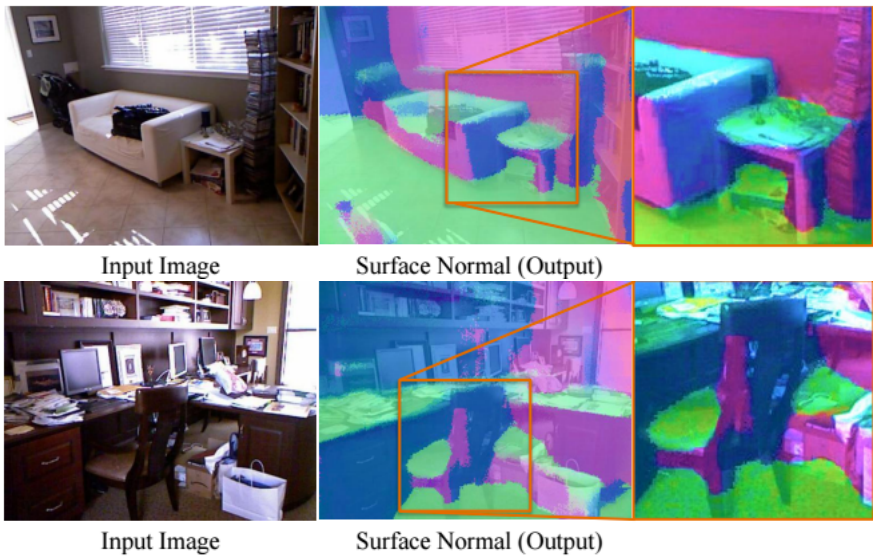
Dense Prediction – Per Pixel

- **Depth/Normal Estimation**

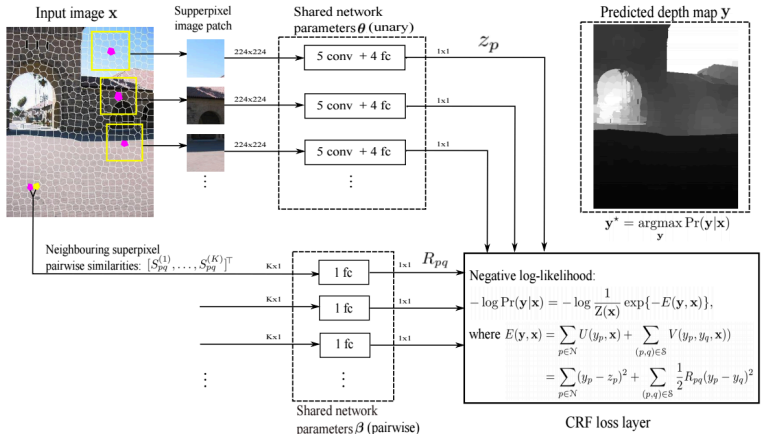


Layer	input	Coarse 1	Coarse 2,3,4	Coarse 5	Coarse 6	Coarse 7	Fine 1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

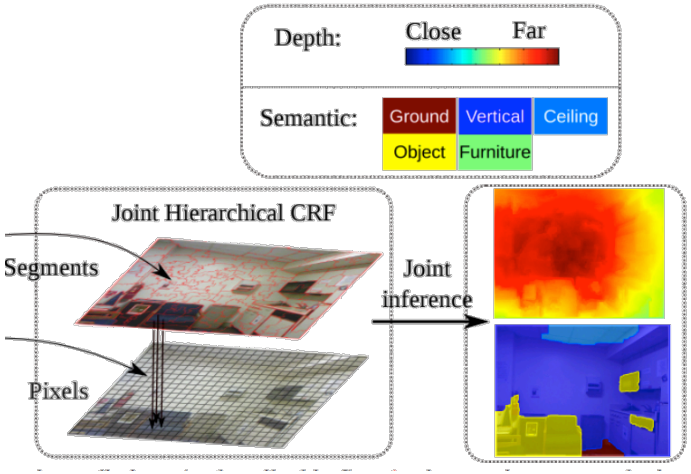
Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. Eigen1et al. NIPS 2014



Designing Deep Networks for Surface Normal Estimation. Wang, et al. CVPR 2015



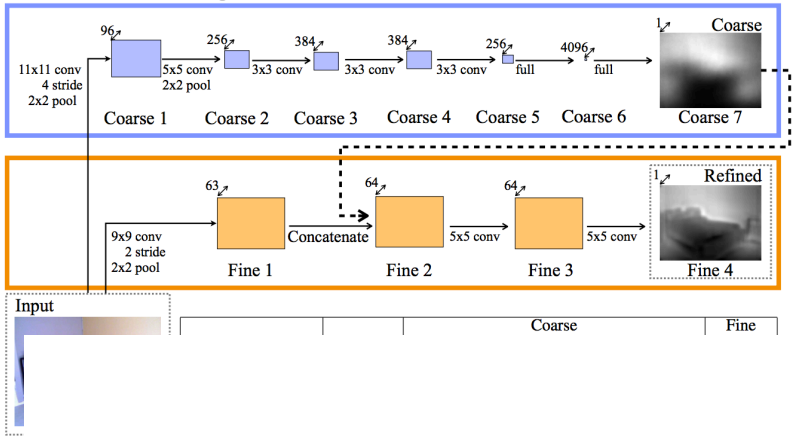
Deep Convolutional Neural Fields for Depth Estimation from a Single Image. Liu et al. CVPR 2015

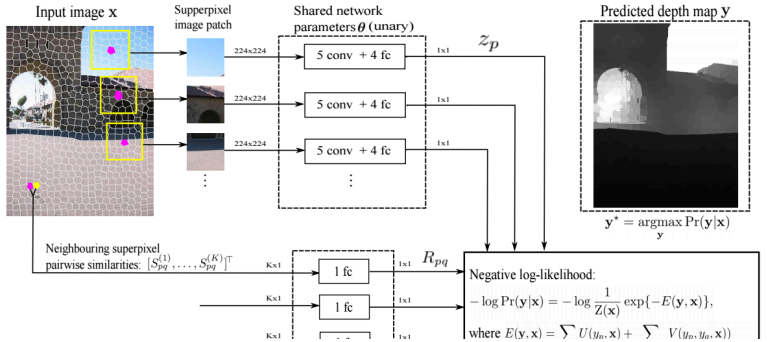


Towards Unified Depth and Semantic Prediction from a Single Image. Wang et al. CVPR 2015

Dense Prediction – Per Pixel

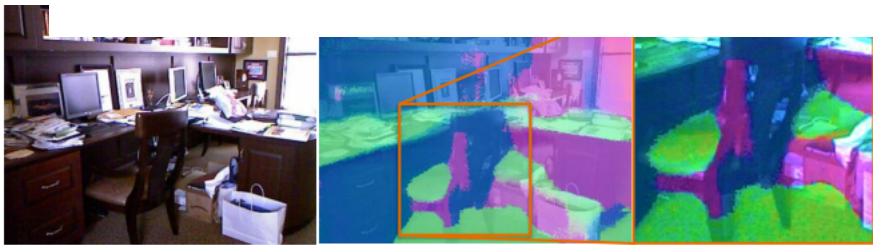
- **Depth/Normal Estimation**



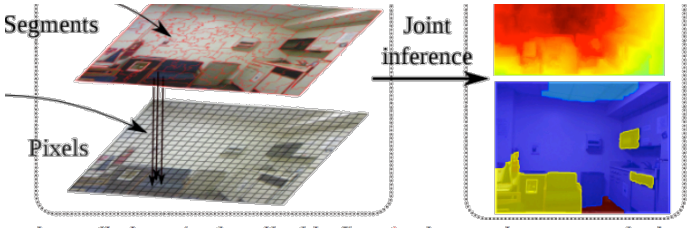


Can CNN handle >1 inputs?

Yes!



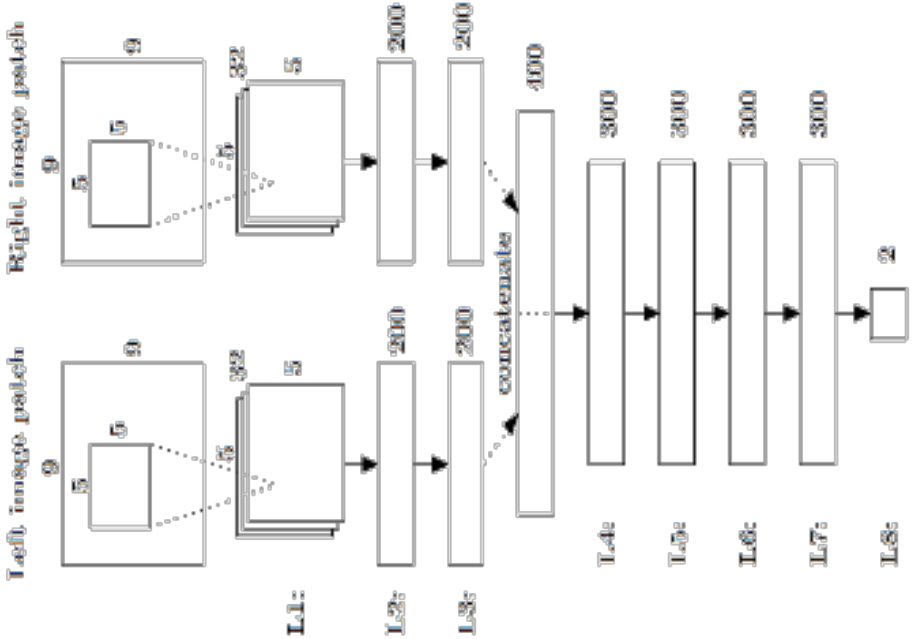
Designing Deep Networks for Surface Normal Estimation.
Wang, et al. CVPR 2015

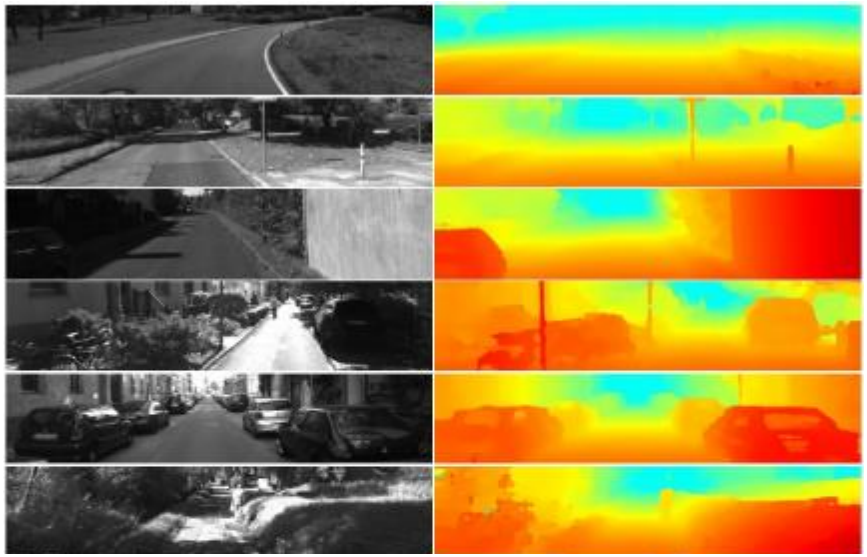


Towards Unified Depth and Semantic Prediction from a Single Image.
Wang et al. CVPR 2015

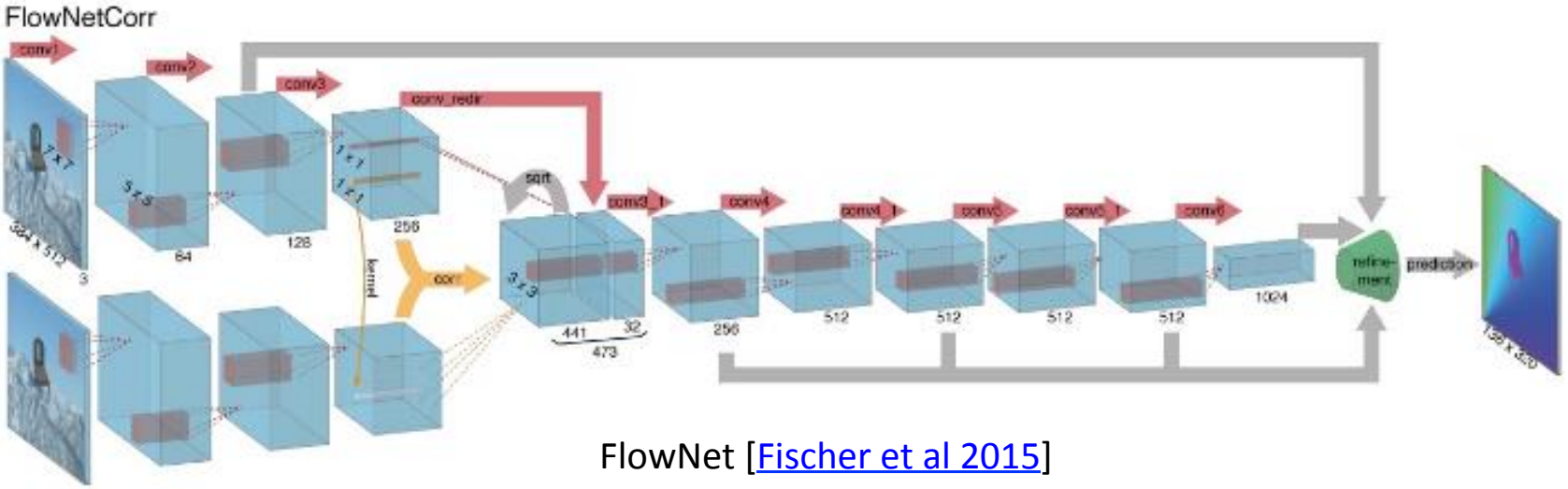
Dense Prediction – Two Streams

- **Correspondences/Matches/Flow**





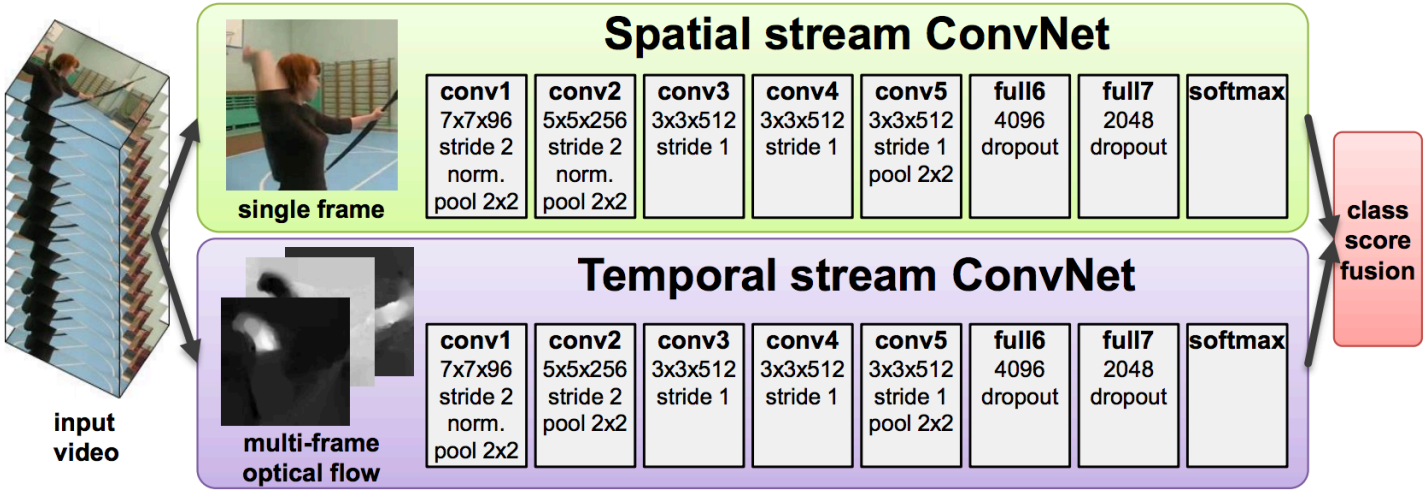
Stereo matching [Zbontar and LeCun CVPR 2015] Compare patch [Zagoruyko and Komodakis 2015]



FlowNet [Fischer et al 2015]

Two Streams

- **Action Recognition**



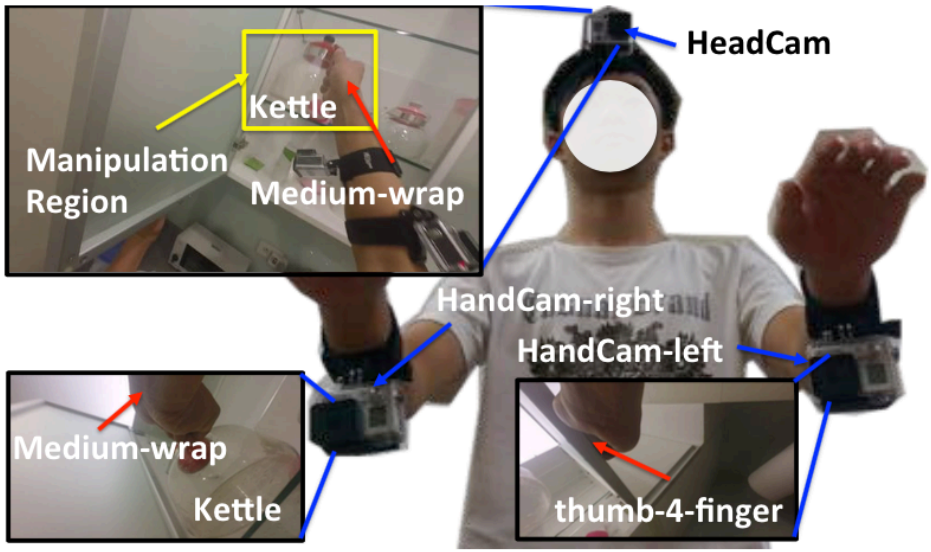
Two-Stream Convolutional Networks for Action Recognition in Videos [Simonyan and Zisserman. NIPS 2014]

- **Cross-view Matching**

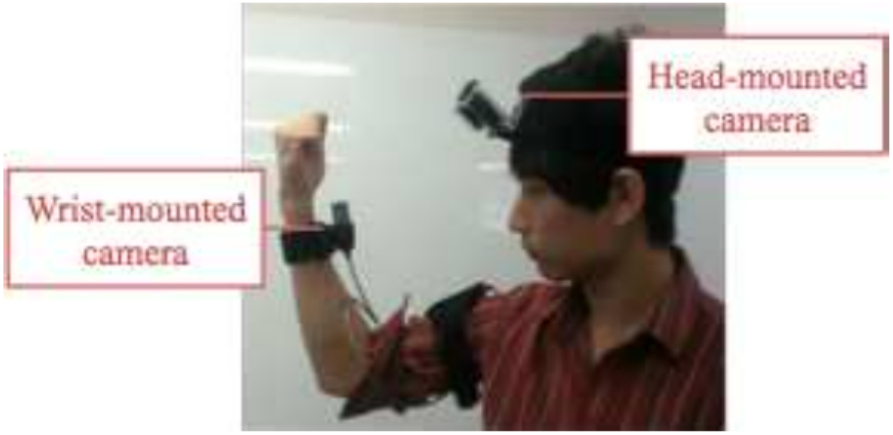
Match ground and aerial images [Lin et al. CVPR 2015]

Two Streams - Wearable Applications

- **Wrist/Hand** Camera



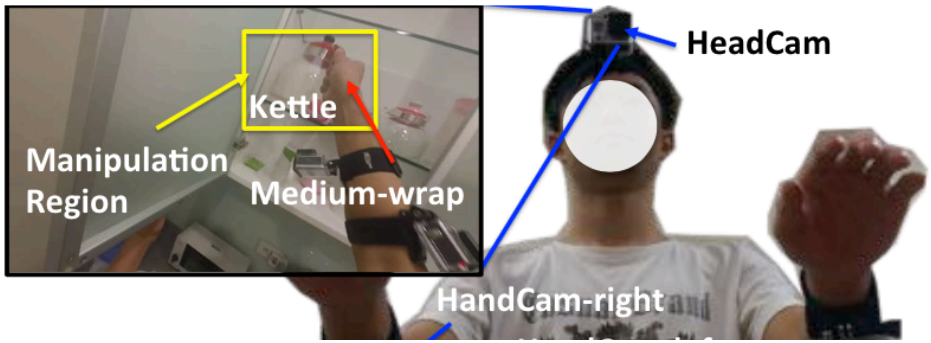
Recognition from Hand Cameras [Chan er al. arxiv:1512.01881v1 Dec. 2015]

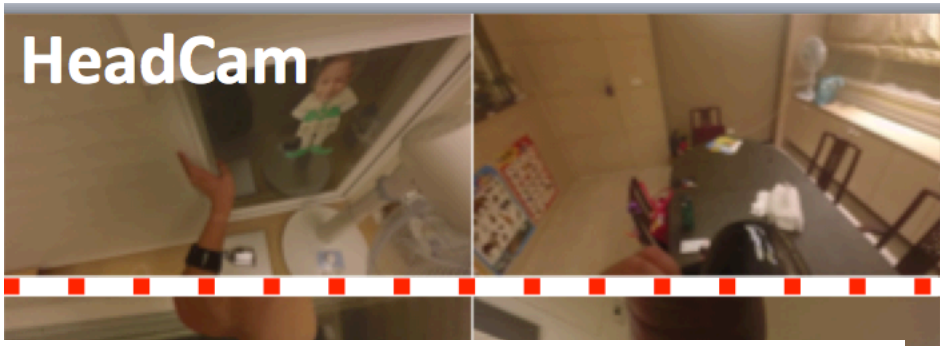


Recognizing Activities of Daily Living with a Wrist-mounted Camera [Ohnishi er al. arXiv:1511.06783v1 Nov. 2015]

Two Streams - Wearable Application

- **Wrist/Hand** Camera





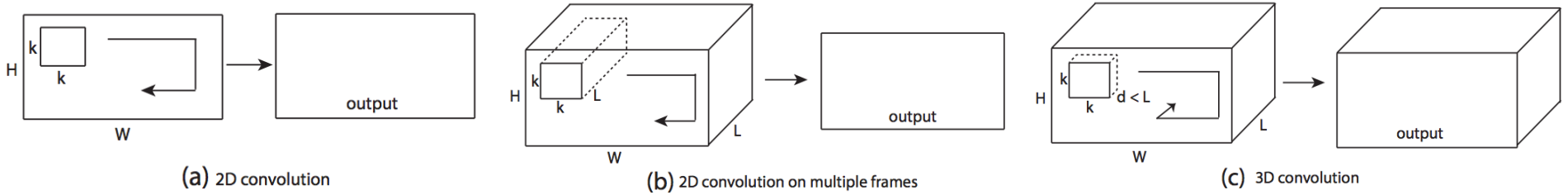
Can CNN handle input dimension > 2?

Yes!

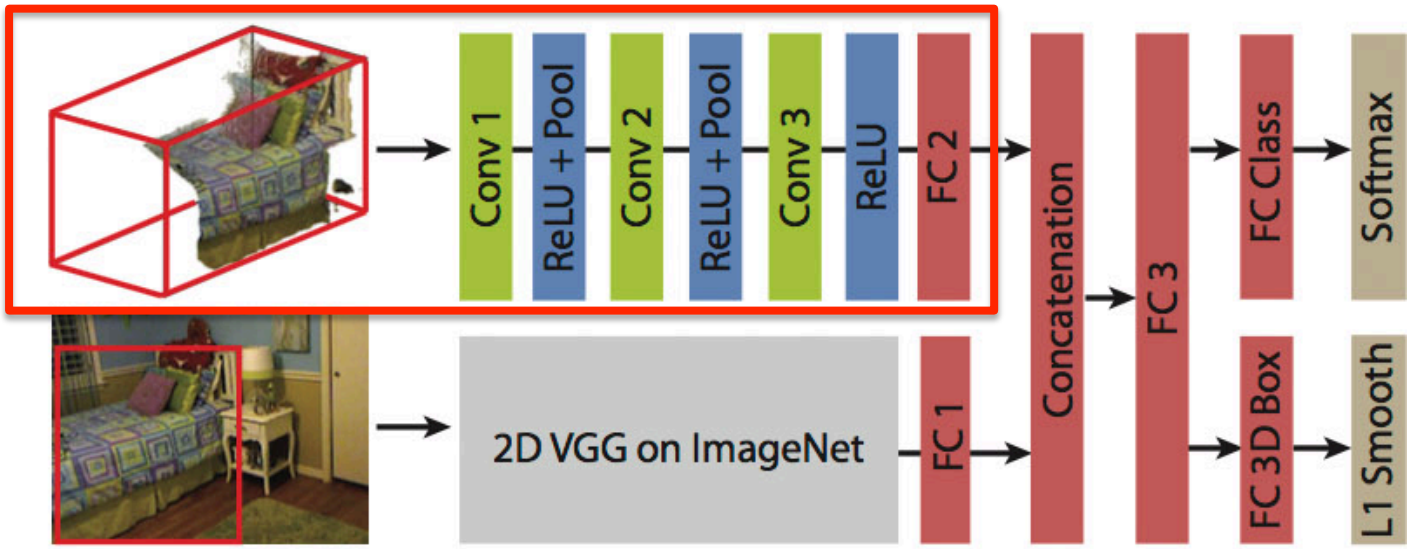
Recognizing Activities of Daily Living with a Wrist-mounted Camera [Ohnishi er al. arXiv:1511.06783v1 Nov. 2015]

Beyond 2D Convolution

- 3D CNN



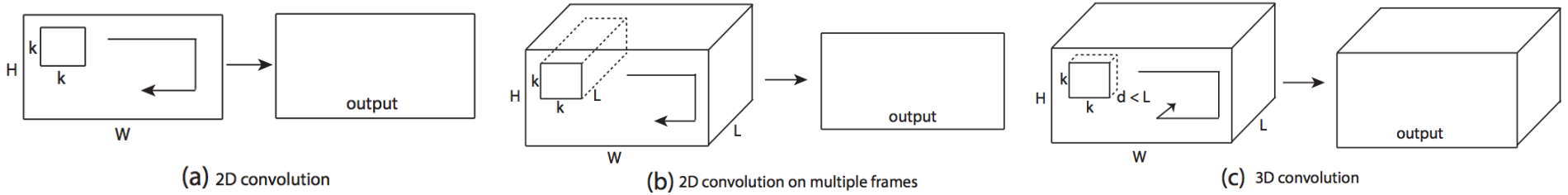
Learning Spatiotemporal Features with 3D Convolutional Networks
[Tran et al. ICCV 2015]
https://github.com/facebook/C3D



Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images
[S. Song, and J. Xiao. arXiv:1511.02300v1 Nov. 2015]
Software Framework: Marvin http://marvin.is

Beyond 2D Convolution

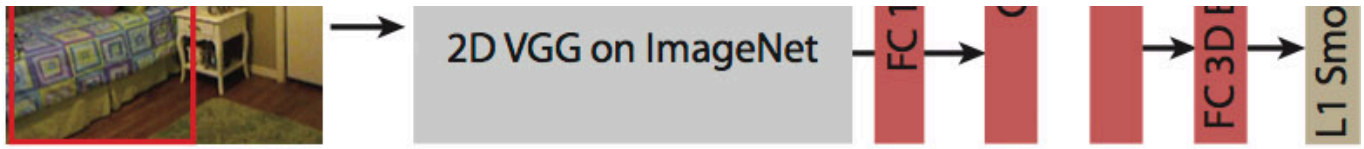
- 3D CNN



Learning Spatiotemporal Features with 3D Convolutional Networks

Can DNN handle Sequence?

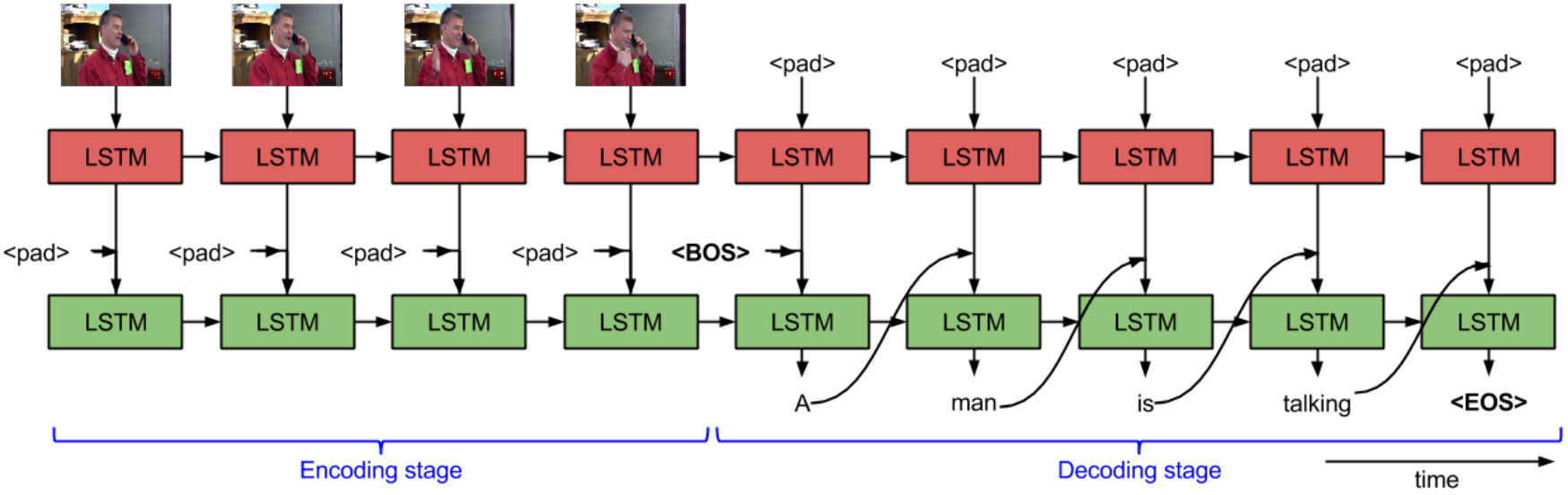
Yes!



Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images
[S. Song, and J. Xiao. arXiv:1511.02300v1 Nov. 2015]
Software Framework: Marvin http://marvin.is

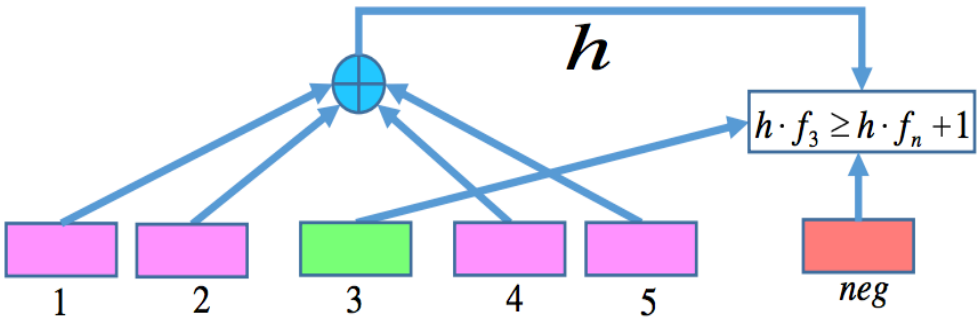
Sequence Modeling

- **Video to Caption**



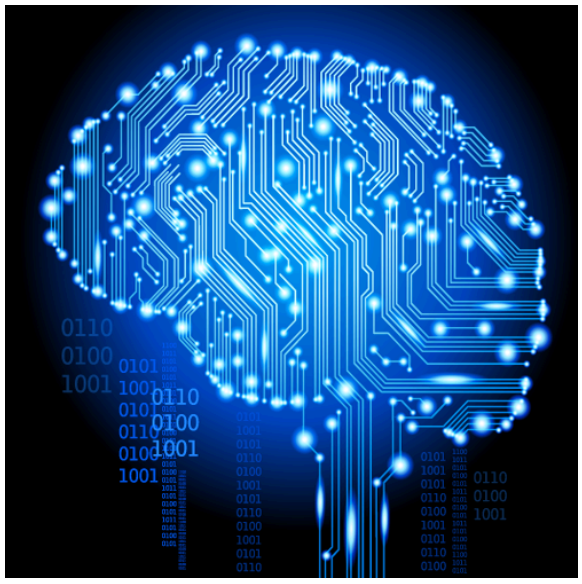
Sequence to Sequence - Video to Text [Venugopalan et al. ICCV 2015]

- **Unsupervised Sequence Modeling**



Learning temporal embeddings for complex video analysis [Ramanathan et al. ICCV 2015]

iPhone 3g Commercial in 2009:
"There's an app for that”



In 2015
"There's a DeepNet for that”

Tools

- Caffe (c++,matlab,python)
- cuda-convnet2 (c++,python)
- Theano (python)
- Torch (lua)
- MatConvNet (matlab)
- Pylearn2 (python)
- TensorFlow (python, c++)
- Marvin – N-dimension (c++,python,matlab)
- MxNet (c++,python,R,js,etc.)

Modified from
Jia-Bin Huang

Resources

- http://deeplearning.net/
  - Hub to many other deep learning resources
- https://github.com/ChristosChristofidis/awesome-deep-learning
  - A resource collection deep learning
- https://github.com/kjw0612/awesome-deep-vision
  - A resource collection deep learning for computer vision
- http://cs231n.stanford.edu/syllabus.html
  - Nice course on CNN for visual recognition
- http://gitxiv.com/category/deeplearning
  - Everything: paper, code, data, community

Slides from
Jia-Bin Huang